MIT-3944-2
MITNE-97

GAMANL, A COMPUTER PROGRAM APPLYING FOURIER TRANSFORMS TO THE ANALYSIS OF GAMMA SPECTRAL DATA

T. Harper, T. Inouye and N. C. Rasmussen

Department of Nuclear Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

August 1968

Contract AT(30-1)-3944
United States Atomic Energy Commission

Safdar Ahmad Ali. Aug 22, 1969

MIT-3944-2
MITNE-97

GAMANL, A Computer Program Applying Fourier Transforms To The Analysis of Gamma Spectral Data

T. Harper, T. Inouye* and N. C. Rasmussen

August 1968

Massachusetts Institute of Technology
Department of Nuclear Engineering

Contract AT(30-1)-3944
United States Atomic Energy Commission

*Work done while on leave from Tokyo Shibaura (TOSHIBA) Electric Company, Ltd., Japan

GAMANL, A COMPUTER PROGRAM APPLYING FOURIER TRANSFORMS TO THE ANALYSIS OF GAMMA SPECTRAL DATA

by

T. Harper, T. Inouye*, and N. C. Rasmussen

Abstract

GAMANL, a computer code for automatically identifying the peaks in a complex spectra and determining their centers and areas, is described. The principal feature of the method is a data smoothing technique employing Fourier transforms. The smoothing eliminates most of the random fluctuations without effecting the spectral resolution and makes identification of maxima using a zero slope criterion possible. Using the same Fourier transform with different constants it is possible with a second transformation to improve the spectral resolution. The computer program has been written in FORTRAN IV for the M.I.T. IBM 360 model 65 computer and also for the Toshiba Electric Company G.E. 635 computer. The complete analysis of a 4096 channel spectrum containing one hundred twenty peaks requires about 75 seconds of computation time.

*Work done while on leave from Tokyo Shibaura (TOSHIBA) Electric Company, Ltd., Japan.

TABLE OF CONTENTS

Figure Captions

Figure 1 The absolute value of the Fourier transforms for two different 4096 channel γ-ray spectra. The energy width of a channel was changed as indicated. The solid lines indicate the approximate shape of the two components of $|F(\omega)|$.

Figure 2 The smoothing filter function $P(\omega)$ showing the shape for 3 different cutoffs used for the runs plotted in Fig. 3. A Gaussian shape with a σ of 128 $(\frac{2\pi}{4096})$ radian/channel is used to reduce function from unity to zero.

Figure 3 The effects of the three different filter functions of Fig. 2 on a doublet taken from the 0.971 keV/channel data. Curve (a) is the original data, curves (b, c and d) result when filter functions (b, c, d) are used. The four curves are arbitrarily displaced by 10^3 counts for clarity of presentation.

Figure 4 The Fourier transform of a single peak and an unresolved doublet from the 0.724-keV/channel spectrum.

Figure 5 The upper curve $|G(\omega)|$ is the absolute value of Fourier transform of the smoothed background subtracted data of the 0.724-keV/channel spectrum. The lower curve $W(\omega)$ is the resolution improvement filter function used to produce Fig. 8. $1/H(\omega)$ is the inverse of the transform of a single spectral peak.

Figure 6 The observed data of an unresolved doublet from a

Figure Captions (Continued)

4096 channel spectrum having a 0.724-keV channel width.

Figure 7 The doublet of Fig. 6 following smoothing and background subtraction.

Figure 8 The unresolved doublet of Fig. 6 showing the effect of the Fourier transformation using the filter function $W(\omega)$ shown in Fig. 5. The small peaks on either side are spurious peaks introduced by the transformation process.

Figure 9 GAMANL Flow Diagram

1. Introduction

This paper describes a computer code employing Fourier transforms for analyzing complex γ-ray spectra such as those obtained with a Ge(Li) detector, multichannel analyzer system. The code was originally developed to analyze the capture γ-ray spectra obtained with the M.I.T. triple coincidence spectrometer (1), but has since been used successfully on a variety of other spectral data. The object of the program is to automatically locate all the spectral peaks and determine their centers and areas. This problem has existed for a long time in spectroscopy and a number of methods for doing all or part of these operations have been proposed (2,3). Most of them were developed for NaI spectra where the total number of peaks was small and the peaks were identified by hand. In a Ge(Li) spectra there are often more than a hundred peaks so it becomes important to have a fast method for automatically identifying each peak. One method for doing this described by Mariscotti (4) uses a second difference to locate the peak. Our experience with second differences indicated that they tended to obscure weak peaks. Mariscotti has handled this problem by doing a multipoint smoothing of the second difference. In the approach described below we have chosen to smooth the original data by a Fourier transform method. Although the Fourier method was found to be a quick accurate way to locate the peaks, we had no runs on similar data using the Mariscotti code which would give a meaningful comparison of the two methods.

Section 2 of this paper describes the theory underlying the Fourier transform method. Section 3 indicates how the theory is applied to the actual data and shows typical results. Section 4 describes the computer code GAMANL. The appendices give the program list, input and partial output.

2. Theory

The method described here is an improved version of one we described earlier (5). It can be divided into four steps: 1) data smoothing, 2) background subtraction, 3) resolution improvement, and 4) energy and intensity determination of the peaks.

2.1 Data Smoothing

The smoothing of the observed data is accomplished by a Fourier analysis technique similar to that often used to separate the signal from the noise in communication theory. In communication theory a function of time is transformed into frequency space, multiplied by an appropriate filter function and then transformed back into time space. In the present case the original data is a function of energy or more exactly a function of channel number so the transformation is into inverse channel number space. By analogy we shall call this "energy frequency space", and use the symbol ω which has units of radians/channel for the variable.

To describe the method mathematically, let the observed data f(E) be represented as the sum of two components

$$f(E) = s(E) + n(E) \qquad (1)$$

where s(E) is the true spectral information and n(E) is the noise which in this case is due principally to random fluctuations in the number of counts in a channel. The Fourier transform of f(E) denoted $F(\omega)$ can be written in the usual notation as

$$F(\omega) = \int_{-\infty}^{\infty} f(E)\, e^{-i\omega E}\, dE \qquad (2)$$

or

$$F(\omega) = S(\omega) + N(\omega) \qquad (3)$$

where $S(\omega)$ and $N(\omega)$ stand for the Fourier transforms of the components of $f(E)$.

The success of the method depends upon $S(\omega)$ and $N(\omega)$ being different functions so that a filter function can be chosen which will eliminate at least part of $N(\omega)$ without seriously affecting $S(\omega)$. Fortunately this is true in most spectra since the spectral peaks are spread over a number of channels and so $S(\omega)$ is made up principally of low frequencies. The noise on the other hand is channel to channel fluctuations and so $N(\omega)$ contains many higher frequencies. Thus there will be a significant difference in $N(\omega)$ and $S(\omega)$ for cases when the spectral peaks are several or more channels wide and the method will be applicable.

Let us define the inverse transform

$$s(E) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, P(\omega)\, e^{i\omega E}\, d\omega \qquad (4)$$

where $P(\omega)$ is the filter function and $s(E)$ will be a smoothed version of the original spectrum. The method of choosing $P(\omega)$ is discussed later. For simplicity we have expressed all the transforms in their integral form; however, when the technique is applied to discrete data the transformations must be used in their discrete form.

2.2 Background Subtraction

In most cases the spectral peak sits on a background which must be subtracted in order to accurately determine the peak area. This background is the result of a number of processes in the source and in the detector and it often can not be expressed accurately

analytically. We have found that to a good approximation the background can be represented as a slowly varying function which connects all the minima in the smoothed data s(E). Care must be exercised in applying this definition, however, since the minima which occur in the case of partially resolved peaks must be excluded. This is accomplished by setting a maximum value for the slope of the background. When the slope connecting two successive minima exceeds this maximum value the minima is ignored and the background line is connected to the next minima which will give an acceptable slope. The smoothed background subtracted data is designated g(E).

2.3 Resolution Improvement

In complicated spectra such as those from (n,γ) reactions there are a number of cases where lines are only partially resolved. In order to determine the peak centers accurately, it would be helpful to have a higher energy resolution. Because of our knowledge of the response of the detector to a monoenergetic γ ray the mathematical limit of resolution is somewhat better than the apparent limit usually expressed as the FWHM (full width of half maximum) of a peak. The detailed theory underlying this method has already been described in the literature by Inouye (6) where it was applied to NaI spectra. Here we briefly restate the results and show their applicability to Ge(Li) spectra as well. To understand the method let us consider Eq. 5.

$$g(E) = \int_{-\infty}^{\infty} h(E-E')\, j(E')\, dE' \qquad (5)$$

where g(E) is the smoothed background subtracted data which can be expressed analytically as the integral where h(E-E') is the response function of the detector and j(E') is the incident spectra which in this case may

be considered to be a series of δ functions in energy as expressed in Eq. 6.

$$j(E') = \sum_i A_i \, \delta(E'-E_i) \; . \tag{6}$$

Since Eq. 5 is a convolution integral, its transform can be expressed as

$$G(\omega) = H(\omega) \; J(\omega) \; . \tag{7}$$

Now since we can experimentally determine g(E) and h(E'-E) we can calculate $G(\omega)$ and $H(\omega)$ and can therefore determine j(E), the incident spectrum as shown by Eq. 8.

$$\begin{aligned} j(E) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega)/H(\omega) \; e^{i\omega E} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) \; W(\omega) \; e^{i\omega E} d\omega \end{aligned} \tag{8}$$

where $W(\omega) = \frac{1}{H(\omega)}$ is the resolution improvement filter function. The problem in practice is that g(E) has some noise in it since the smoothing processes eliminate only part of noise n(E) and the background subtraction also introduces some error. Thus there is no function $W(\omega)$ which will exactly reproduce the input δ functions. Nevertheless, as will be evident in the next section, it is possible to obtain a significant increase in the apparent resolution using this procedure. The principal thing which limits this procedure is the statistical accuracy of the original data.

2.4 Peak Selection and Intensity Determination

The peak selection is accomplished by identifying each maximum in the smoothed background subtracted data g(E). The peak center is obtained by finding the point of

zero slope of a 2nd order fit to the top of the peak. The accuracy of this procedure was checked by comparing the results for a number of peaks to those obtained by finding the centroid of a Gaussian which had been fit to the peak by a least squares method. The results agreed within $\pm$ 0.1 channels on peaks with a σ = 1.5 channels. In terms of energy this was an accuracy of about $\pm$ 0.2 keV and was considerably better than the overall reproducibility of about 0.7 keV of typical runs. However, where greater accuracy is required more sophisticated peak fitting methods (e.g., Ref. (3)) can be used. The peak center in channel number is converted to energy using calibration lines of known energy and a correction is made for the small nonlinearities of the system.

To determine the intensity of a γ ray it is necessary to know the peak area and both the intrinsic and geometric efficiency of the system. In addition, in the capture γ-ray work it was desired to convert this intensity into the number of γ rays per 100 captures. The number of captures was calculated by the code using a measured incident flux and the known cross section.

The area under a spectral peak was determined by two different methods. The first method simply added up all the counts between successive zeros in the smoothed background subtracted data. As noted earlier, cases of doublets and triplets were identified and treated separately. In a majority of the cases this method worked very well. However, occasionally a peak will have a small bump in its tail that is not picked up by the doublet identification method. When this happens in small peaks it can lead to a large error in peak area. To check this a second method of peak area determination developed by Hamawi (7) was added to the program.

The second method takes advantage of the fact that we know that spectral peaks are very nearly true Gaussians, and further, from a careful study of strong isolated peaks in the spectrum, the standard deviation as a function of energy can be determined. The peak area is then determined by

finding the peak center as before. Using the height of the peak and its standard deviation, the area is calculated.

3. Application of Method

3.1 Choice of Smoothing Filter Function

To determine the proper smoothing filter function it is useful to plot $|F(\omega)|$ vs. ω. Such a plot for two typical 4096 channel spectra are shown in Fig. 1. The actual transformation used gives one ω point for each energy point. For clarity of presentation we have actually plotted only every fiftieth point in ω space. The x's represent a run where the gain was adjusted to produce a channel width of 0.971 keV. The circles represent a lower gain which gave a 2.063-keV channel spacing. The approximate shape of the components of $F(\omega)$ are shown as the solid lines. As expected in the 0.971-keV run where the peaks contain more channels the signal component $S(\omega)$ contains less high frequencies. The filter function must be of a form that passes frequencies below the break in the curve but eliminates those frequencies above the break where the noise component $N(\omega)$ dominates. A filter function shape that was found to work well is shown in Fig. 2. The three different curves represent functions with different cutoffs. A Gaussian shape used to reduce the curve from unity to zero has a $\sigma = 128$. It is important that the filter function be a smoothly varying function in order that oscillations not be introduced in the inverse transformation. To illustrate the effect of changing the cutoff frequency that data used to obtain the 0.971 keV/channel curve of Fig. 1 was processed using the three different filter functions shown in Fig. 2. A portion of the 4096 channel spectrum showing the effect of each of these filter functions on a doublet is shown in Fig. 3. The cutoff that begins at 412 $(\frac{2\pi}{4096})$ rad/channel is the one predicted from Fig. 1 and it produces considerable smoothing action without decreasing the peak to valley ratio in the doublet. For clarity the curves were arbitrarily shifted by 1000 counts/

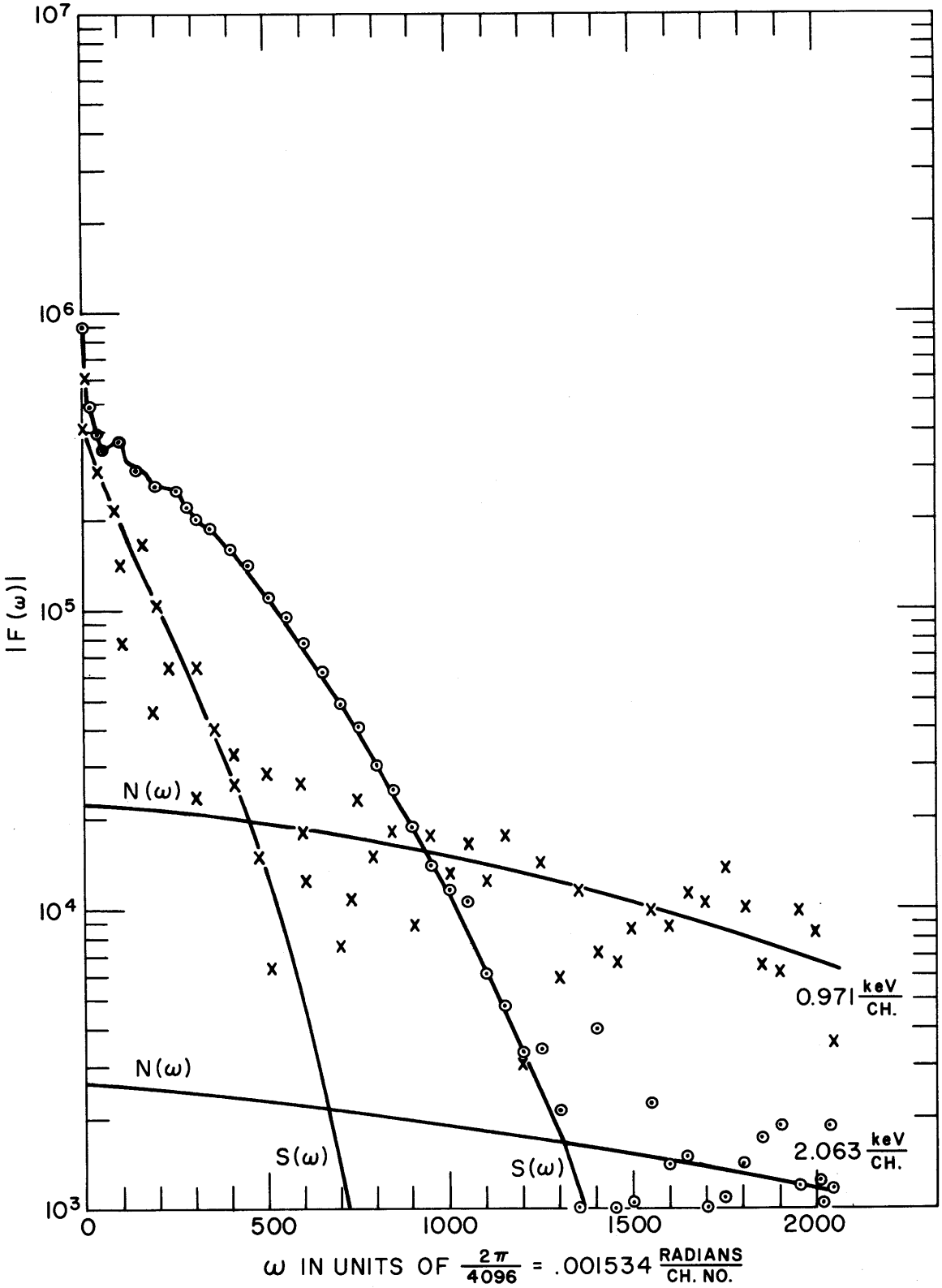

10^7
10^6
10^5
10^4
10^3
|F(ω)|
N(ω)
N(ω)
S(ω)
S(ω)
0.971 keV/CH.
2.063 keV/CH.
0
500
1000
1500
2000
ω IN UNITS OF 2π/4096 = .001534 RADIANS/CH. NO.

Figure 1

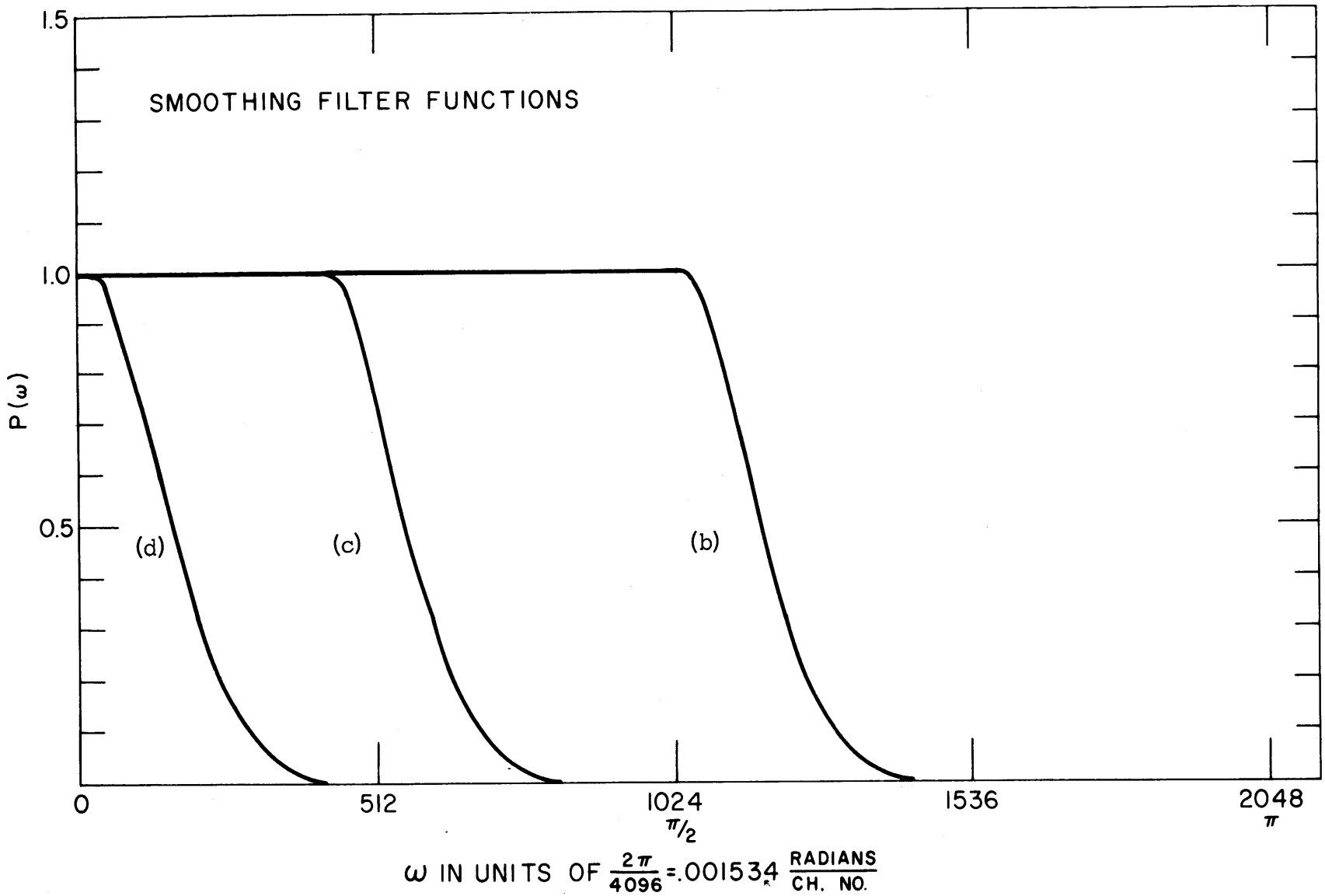

SMOOTHING FILTER FUNCTIONS
1.5
1.0
0.5
P(ω)
(d)
(c)
(b)
0
512
1024
π/2
1536
2048
π
ω IN UNITS OF 2π/4096 =.001534 RADIANS/CH. NO.

Figure 2

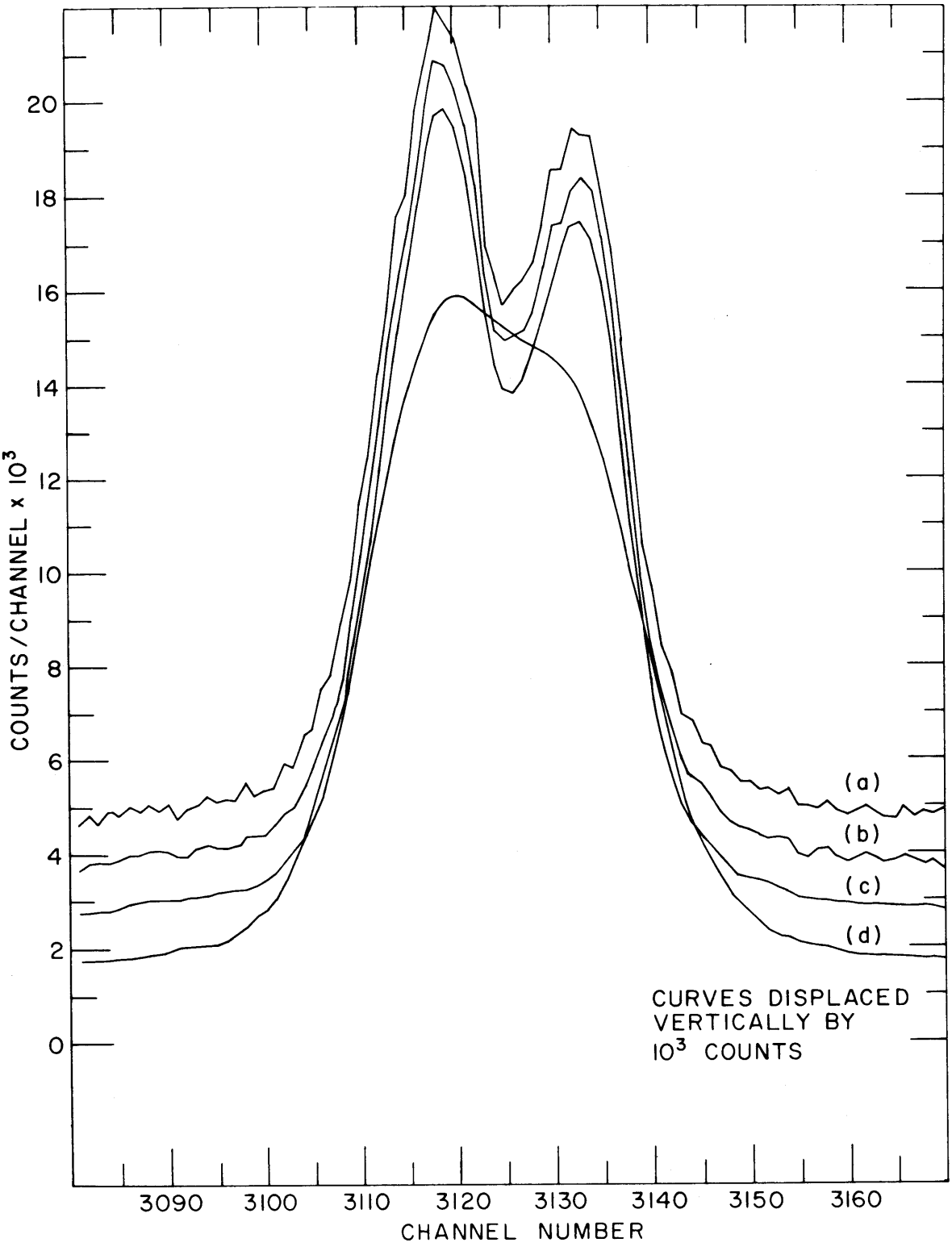

COUNTS/CHANNEL x 10^3
20
18
16
14
12
10
8
6
4
2
0
3090
3100
3110
3120
3130
3140
3150
3160
CHANNEL NUMBER
(a)
(b)
(c)
(d)
CURVES DISPLACED
VERTICALLY BY
10^3 COUNTS

Figure 3

channel. Careful analysis also revealed that the peak centers were not shifted by the smoothing process. In addition the smoothing action is rather insensitive to both the location of the cutoff and σ of the Gaussian. Variations of 10 to 15% in either of these quantities gave appreciably the same results. Thus the same filter function can be used on all runs of approximately the same gain. As expected the function with the higher cutoff does not produce as much smoothing and the very low cutoff produces so much smoothing that considerable spectral information is lost.

3.2 Choice of Resolution Improvement Filter Function

According to the theory discussed above $W(\omega) = 1/H(\omega)$ where $H(\omega)$ is the transform of the response of the system to a unit δ function input. The $H(\omega)$ can be determined by carefully analyzing the response of the system to a single γ ray. Figure 4 shows the transformation for a large single peak and also for an unresolved doublet. The case of the unresolved doublet will be discussed later. The function $1/H(\omega)$ is obtained using the normalized $A \cdot H(\omega)$. In Figure 5 the normalized function $1/H(\omega)$ is plotted for the singlet shown in Fig. 4. The function $|G(\omega)|$ is the transform of the smoothed data for the entire spectrum from which the single line was taken. Its cutoff is different from the previous examples because the gain for this run was set to give 0.724 keV/channel.

Although the theory predicts that $1/H(\omega)$ is the best filter function to use, in practice this is not the case because of the noise. As can be seen the function $1/H(\omega)$ increases without limit as the frequency increases. This would create no problem in the absence of noise since the signal tends toward zero equally rapidly but when the noise is multiplied by this very large value it produces many artificial peaks when the inverse transform is carried out. To eliminate this problem we have found that it is necessary

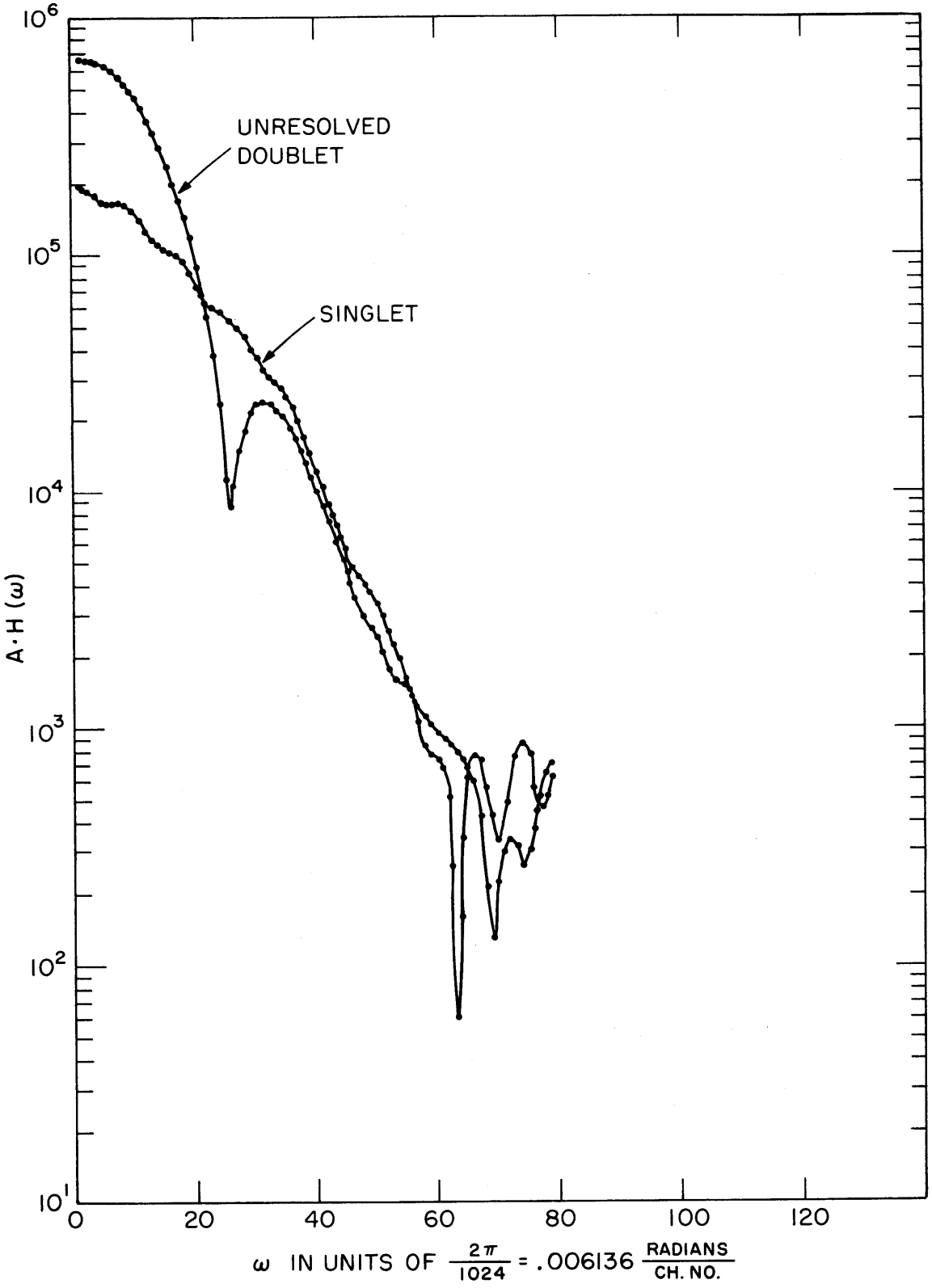

UNRESOLVED
DOUBLET
SINGLET
A·H (ω)
10^6
10^5
10^4
10^3
10^2
10^1
0
20
40
60
80
100
120
ω IN UNITS OF 2π/1024 = .006136 RADIANS/CH. NO.

Figure 4

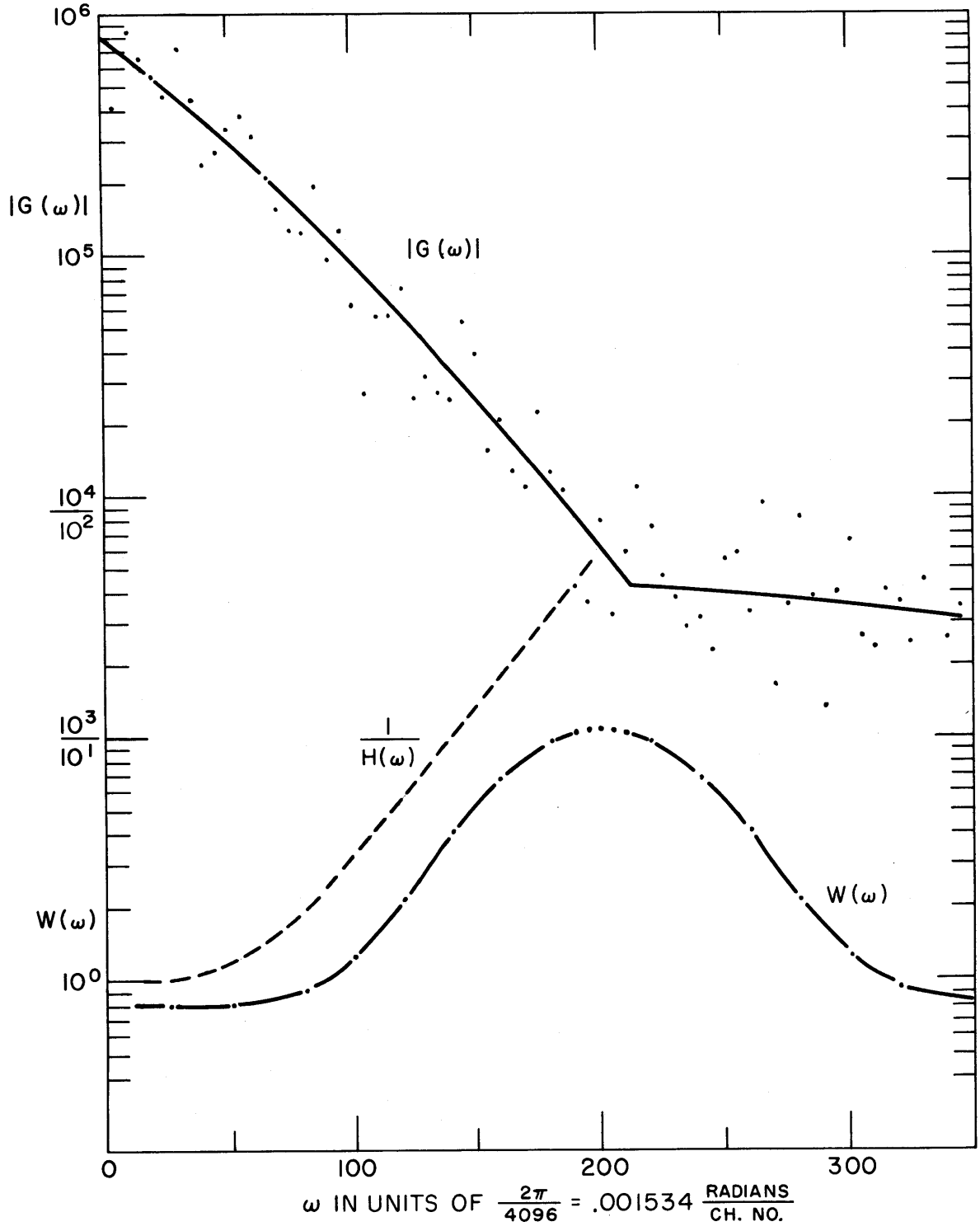

10^6
$|G(\omega)|$
10^5
$|G(\omega)|$
$\frac{10^4}{10^2}$
$\frac{10^3}{10^1}$
$\frac{1}{H(\omega)}$
$W(\omega)$
10^0
$W(\omega)$
0
100
200
300
ω IN UNITS OF $\frac{2\pi}{4096} = .001534 \frac{\text{RADIANS}}{\text{CH. NO.}}$

Figure 5

that $1/H(\omega)$ return to zero at frequencies above the cutoff in $G(\omega)$. By trial and error it has been found that a Gaussian plus a constant is a very satisfactory function for accomplishing the desired result. This function has four adjustable parameters: the center, amplitude, standard deviation of the Gaussian and the constant, all of which must be optimized for the data being analyzed. Once this is done for runs of a given gain it works equally well for subsequent runs at approximately the same gain.

In order to demonstrate the method the doublet of Fig. 3 was recorded under conditions which produced a much worse energy resolution in the system. The original data from this run in the region of the doublet are shown in Fig. 6. Figure 7 shows the results of smoothing and background subtraction as described above. The curve marked $|G(\omega)|$ in Fig. 5 is the transform of the smoothed background subtracted data. This was multiplied by the function $W(\omega)$ and the inverse transformation produced the doublet shown in Fig. 8.

Note that although a dramatic increase in resolution is possible the process introduces small fictitious peaks on either side of the doublet. The result is that the technique is fine for examining a specific doublet to determine its components but it cannot be applied in this drastic a manner to the entire spectrum without introducing a number of small spurious peaks which will interfere with the identification of real, small peaks. In addition the area under a peak is sometimes changed by a spurious peak from a nearby true peak. In practice, therefore, all peaks are identified and their areas, centers and widths determined prior to the second transformation. Then those peaks with widths greater than expected are examined by the above process to locate the position of their components. The amplitude of the components is then determined by adjusting them to best fit the unresolved doublet in the smoothed data.

It has been found that if the function $W(\omega)$ is not properly chosen it can shift the position of the components

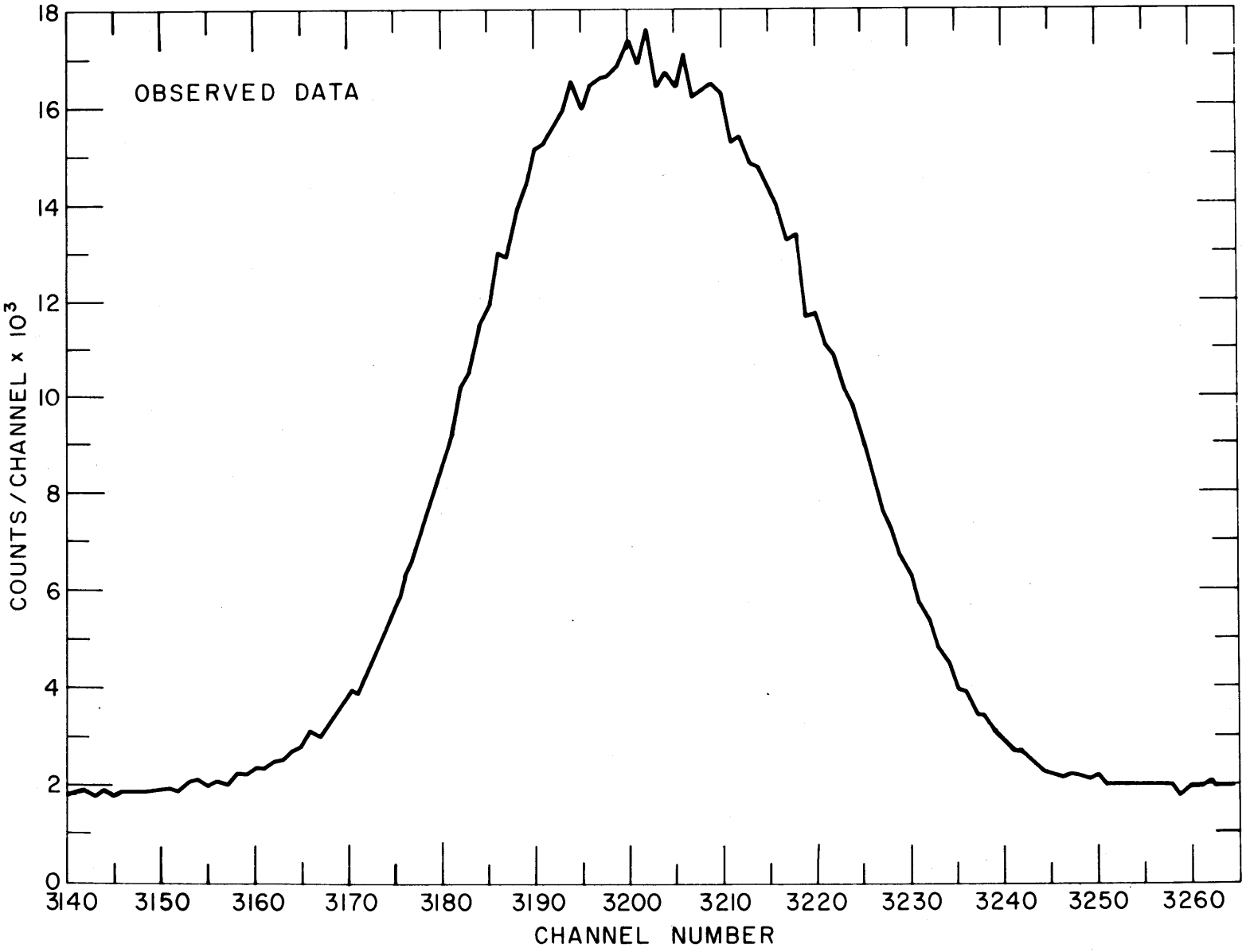

OBSERVED DATA
COUNTS / CHANNEL x 10^3
0
2
4
6
8
10
12
14
16
18
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
CHANNEL NUMBER

Figure 6

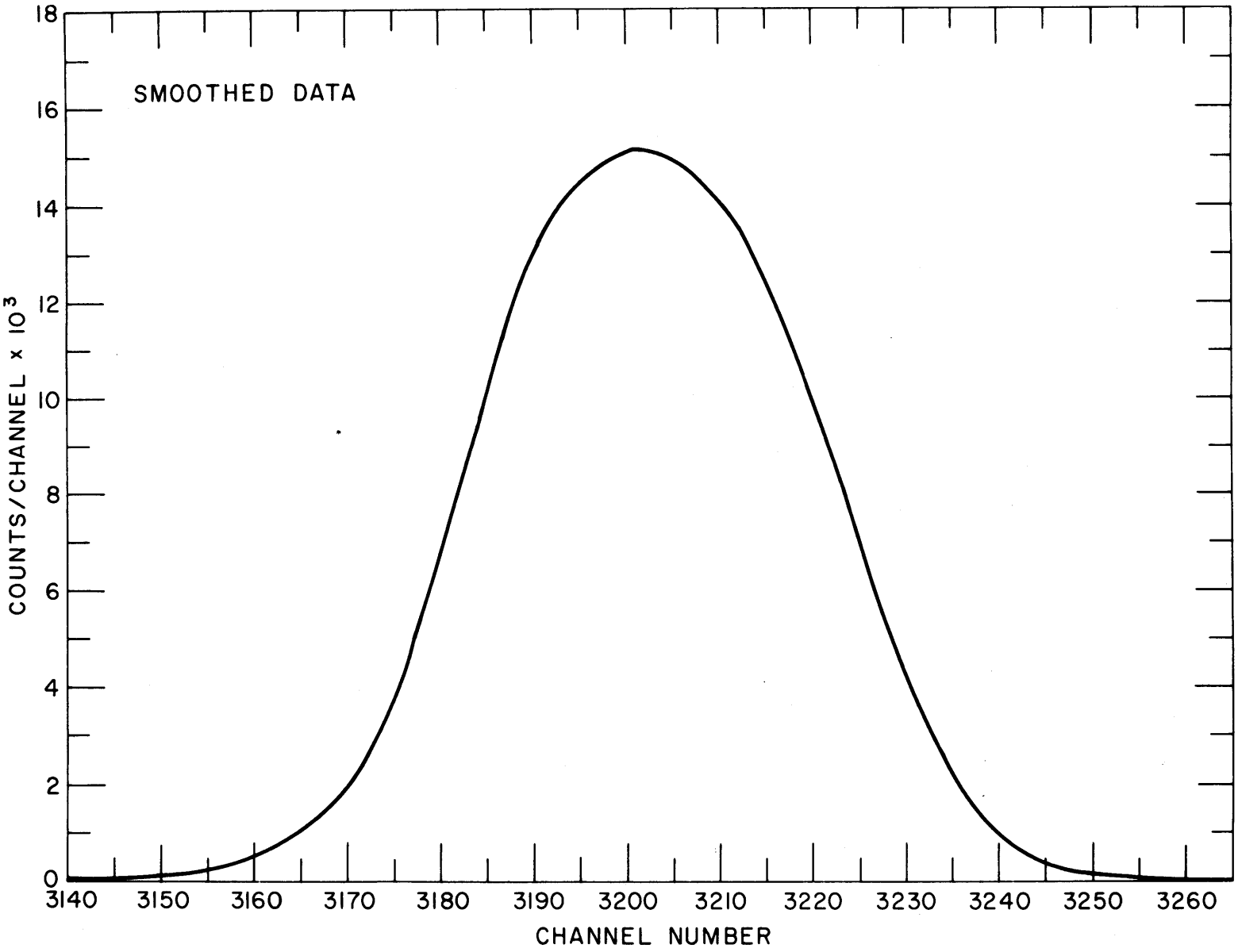

SMOOTHED DATA
COUNTS/CHANNEL x 10³
0
2
4
6
8
10
12
14
16
18
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
CHANNEL NUMBER

Figure 7

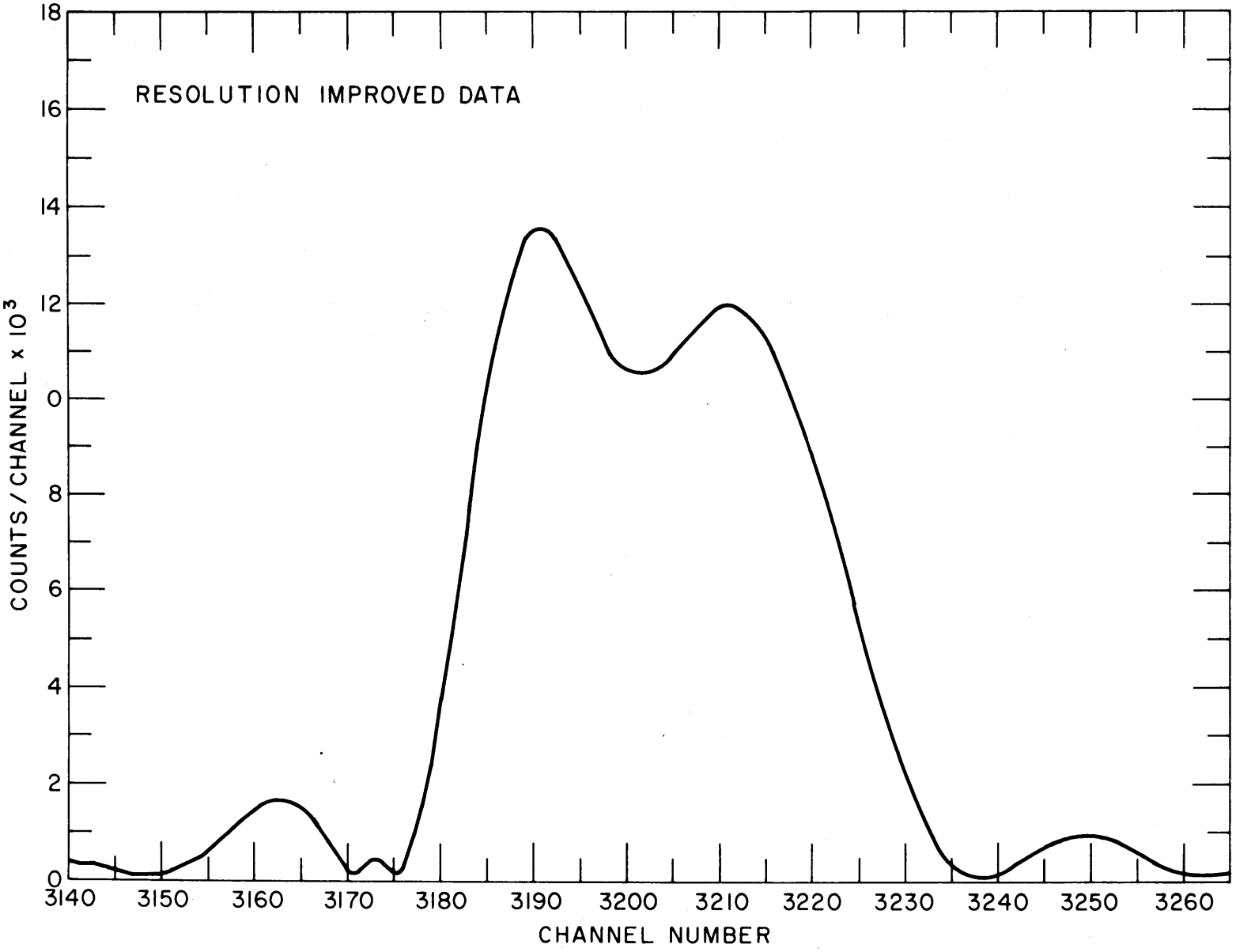

RESOLUTION IMPROVED DATA
COUNTS / CHANNEL x 10³
CHANNEL NUMBER
0
2
4
6
8
0
12
14
16
18
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260

Figure 8

by as much as 1 to 2 channels in data of the type shown here. It is necessary, therefore, to test the function prior to using it. The check that has been used consists of locating the peak centers of the single isolated peaks of the spectrum as described earlier and then performing the transformation using the proposed $W(\omega)$. If there is no resulting shift in the centers of single isolated peaks then the function is assumed to be satisfactory.

It is not always possible to know if the peak is an unresolved doublet. It is possible to check this, however, by looking at the transform of the peak in question. The transform of the unresolved doublet of Fig. 7 is shown in Fig. 4 where it is compared to a singlet peak from the same spectrum. The very noticeable minimum characteristic of doublets is explained by Inouye in Ref. 6.

3.3 Transform Algorithm

A computer code, GAMANL, has been developed for carrying out the gamma spectra analysis discussed above and is described in Section 4. One of the problems encountered with the use of the Fourier Transform equations in their usual form is that N^2 computation operations are required where N is the number of points in energy space, i.e. the number of channels in the spectrum. This means that for N = 4096 the computing time becomes quite long, being on the order of 1/2 hour. Two methods have been used to greatly reduce this computation time. The first method used was to section the data into n smaller segments and to take the transform of each segment separately. This results in $n\left(\frac{N}{n}\right)^2$ operations and reduces the computation time by a factor of $\frac{1}{n}$. For a section length of 16 channels flanked by 10 and 11 channels on each side to correct for end effects, the sectioning method for N = 4096 channels required $256\ (37)^2$ operations or time reduction of a factor of $\frac{256\ (37)^2}{(4096)^2} \simeq \frac{1}{210}$ as compared to the direct method.

The second method used the Fast Fourier Transform (FFT) as developed by Cooley and Tukey (8). The time required by the FFT to transform N energy points is $N(\log_2 N)$, which reduces the computation time by a factor of $\log_2 N/N$ as compared with the usual transformation. Thus for a 4096 channel spectra the time used by the FFT is $\log_2 4096/4096 = 1/256$ of the direct transform. In addition it gives a complete set of Fourier coefficients in ω space useful in the filter function determination. The sectioning does not give these coefficients in as convenient a form.

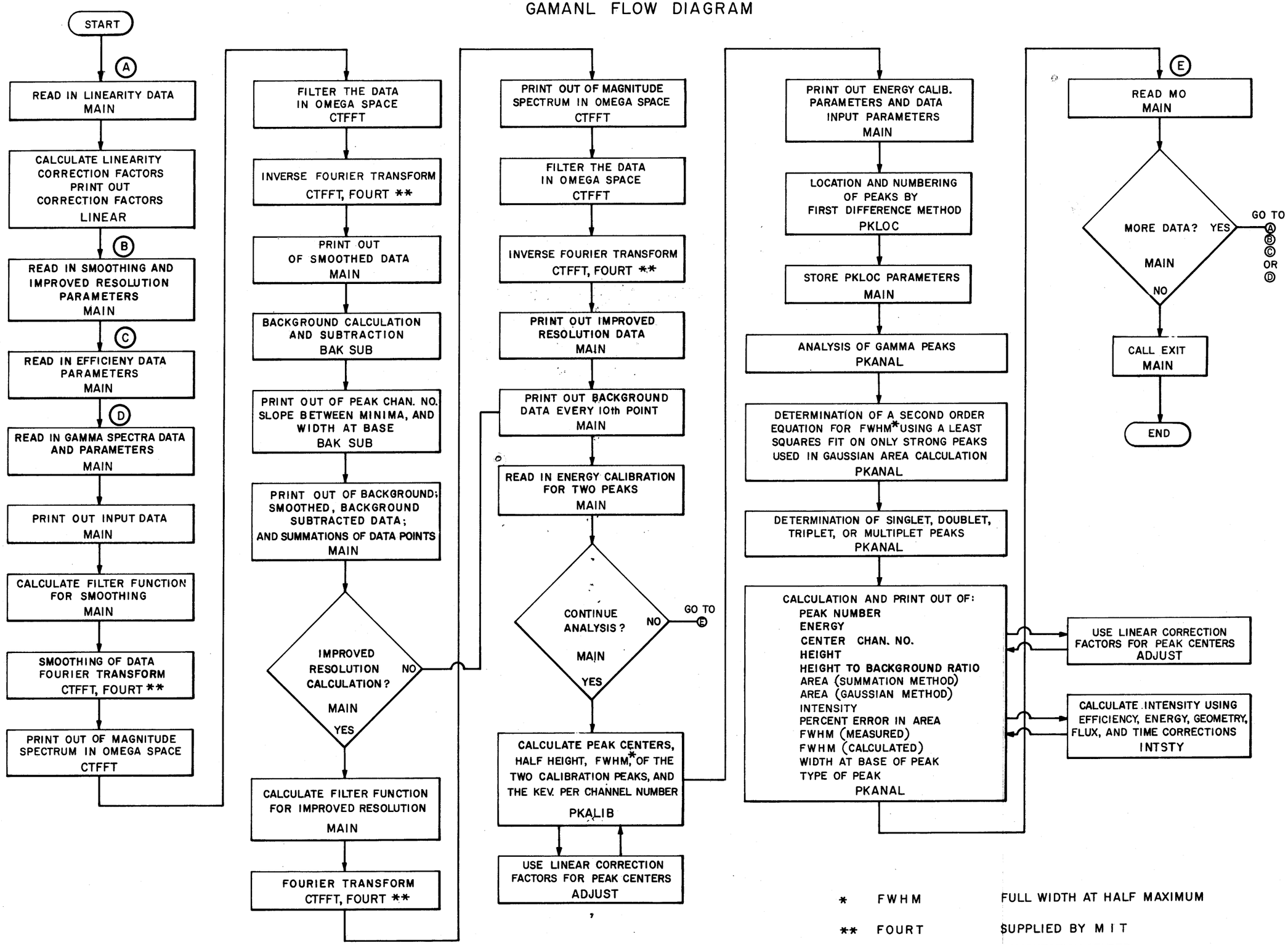
4. GAMANL - Description

The block diagram of GAMANL is given in Fig. 9. The program described here uses the FFT method of carrying out the transformation. The data for calculating the linearity of the multichannel analyzer system is read in first and calculated using subroutine LINEAR. A description of this subroutine is given in Appendix D. These correction factors are stored in the CNTR and CORR array. They are used later in determining the energy of a gamma peak.

The Smoothing and Improved Resolution parameters are then read in along with the efficiency data, run parameters and the gamma spectra input data. The spectra data is printed out and the smoothing filter function (determined by the input parameters) is calculated.

The data is smoothed by the use of the CTFFT subroutine. The Fast Fourier Transform is carried out by FOURT, a subroutine supplied by the MIT Computation Center and written by N. Brenner (9). Subroutine CTFFT performs the actual filtering of the data in which FOURT is used twice, once for the direct transform and again for the inverse transform after the data has been multiplied by the smoothing filter function. The magnitude of the Fourier coefficients in ω space are also printed out in CTFFT.

GAMANL FLOW DIAGRAM



* FWHM — FULL WIDTH AT HALF MAXIMUM

** FOURT — SUPPLIED BY M I T COMPUTATION CENTER

Figure 9

Upon return to the MAIN program the smoothed data is printed out and the background calculation and subtraction are carried out in subroutine BAKSUB, described in Appendix E.

Returning to the MAIN program the background and the smoothed, background-subtracted data are printed out. If no improved resolution is performed the program skips to the printing out of the background for every 10^{th} point. This is done so as to be able to use the array storing the background, TBL, for other purposes such as storing the peak parameters in PKANAL.

If improved resolution is carried out, the improved resolution filter function is calculated using the input parameters. The Fourier Transform of the smoothed, background-subtracted data is calculated using CTFFT and FOURT. The magnitudes of the Fourier Coefficients are printed out. The data in ω space is filtered using the improved resolution filter function. The inverse Fourier Transform is carried out by FOURT and CTFFT. Returning to the MAIN program the improved resolution, background-subtracted, smoothed data is printed out along with the background for every 10^{th} point.

The energy calibration data for two peaks is read in; if the analysis is to be stopped the program transfers to the end of MAIN and stops. If the analysis is to continue subroutine PKALIB is called and the peak center, height and variance of the two energy calibration peaks are calculated. The linearity correction factors as calculated in LINEAR are used in subroutine ADJUST to calculate the corrected peak centers for the calibration lines. From the two calibration peaks the linear relation between channel number (corrected for linearity) and energy is determined.

The calculated calibration data and other data parameters are printed out and subroutine PKLOC is called. PKLOC locates and numbers the peaks using a first difference calculation; and prints out the number of peaks in the

spectra.

Subroutine PKANAL performs the analysis of the spectral peaks. It is broken into two main parts. The first of which is the determination of a second order equation to determine the full width at half-maximum, FWHM, as a function of energy (channel number). A least squares fit is used using only the strong peaks of the spectra. From the equation so determined the FWHM of any peak can be calculated. It is this FWHM that is used in the Gaussian area calculation. The second part analyzes each peak in order of increasing energy. The data from PKLOC are used to determine if the particular peak considered is in singlet, doublet, triplet, or higher multiplet form. Each case is considered separately, while in the higher multiplet case the three strongest peaks are identified and then considered as a triplet case. Certain criteria are applied to the peak being considered, namely that it must have an area error less than a specified maximum value as determined from the input data. If it satisfies this criteria then the peak number, energy, center (channel number), height in counts, height to background ratio, area by summation method, area by Gaussian method, intensity, percent error in area by Gaussian method, FWHM measured, FWHM calculated by 2^{nd} order equation, width at base and multiplet order of peak are calculated and printed out.

The energy and peak center calculation use the linearity correction factors from LINEAR with the calculation being done in subroutine ADJUST. The intensity calculation uses the efficiency versus energy data, flux, cross section, time, and geometry corrections, and is performed by subroutine INTSTY, described in Appendix F.

After completing PKANAL the program returns to MAIN and reads a dummy variable MO. If there is more data, the program returns and reads the new data. If there is no more data the program is ended.

Compilation of the entire program required 50 seconds

on the MIT System 360/65 computer. Execution of a 4096 channel data deck containing approximately 120 peaks with the improved resolution section being used, and thus requiring four transforms of 4096 channels took 74.8 seconds using the compiled machine deck. The final tabular output of PKANAL of this run is shown in Appendix C. An input list to GAMANL is given in Appendix B. The program list of GAMANL in FORTRAN IV is given in Appendix A.

5. Conclusion

The above program has been used at both M.I.T. and the Central Research Laboratory of Toshiba Electric Company on a wide variety of γ-ray spectra (10). In all these applications it has been found to be a fast, accurate method for automatically locating peaks and determining their centers and areas. As noted the smoothing allows the use of approximate methods for determining peak centers and the background function without serious loss of accuracy. In some very precise applications it may be desirable to use more exact methods for this but for the spectra studied the accuracy of the approximate methods was acceptable. The program is available in FORTRAN IV.

APPENDIX A

List of GAMANL, gamma analysis program, by subroutine. Subroutine FOURT was supplied by the M.I.T. Computation Center.

```
            C     DATA ANALYSIS USING ORIGINAL DATA
 0001               COMPLEX DATA
 0002               COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                   1,CORR(200),EFFCY(100)
 0003               COMMON IPUNCH,JREAD,JPRINT,JPUNCH
            C
 0004               JREAD = 5
 0005               JPRINT = 6
 0006               JPUNCH = 7
            C     LINEARITY CALCULATION
 0007             2001 CONTINUE
 0008 (1) ---->     READ (JREAD,200) LR,NOCHAN,N2,N3,XN,LIN
 0009               200 FORMAT (4I5,F5.0,I5)
            C     LR       LINEARITY RUN NUMBER.
            C     NOCHAN   NUMBER OF CHANNELS IN THE LINEARITY RUN, NOCHAN LEQ 4095.
            C     N2       LOW CHANNEL PEAK NUMBER USED IN LINEARITY CALC.
            C     N3       HIGH CHANNEL PEAK NUMBER USED IN LINEARITY CALC.
            C     XN       BACKGROUND SUBTRACTION LIMIT USED FOR LINEARITY CALC.
            C     LIN LEQ ZERO USES PREVIOUSLY CALC. LINEARITY DATA.
            C     IF LIN LEQ ZERO NOCHAN IS THE NUMBER OF PEAKS IN CNTR AND CORR
 0010               IF (LIN) 131,131,101
 0011             101 CONTINUE
 0012 ---->         READ (JREAD,100) (TBL(I),I=1,NOCHAN)
 0013 (2)           100 FORMAT (7X,7(F6.0,1X)/(8(F6.0,1X)))
            C     TBL      LINEARITY DATA ARRAY
            C     SUBROUTINE LINEAR CALCULATES THE CENTERS AND CORRECTION FACTORS
            C     FOR THE LINEARITY DATA PEAKS.
 0014               CALL LINEAR (LR,N2,N3,XN,NOCHAN)
 0015               GO TO 2002
 0016             131 CONTINUE
 0017               WRITE  (JPRINT,152)
 0018             152 FORMAT ('1',14X,' CNTR ARRAY',5X,' CORR ARRAY')
 0019               DO 132 M=1,NOCHAN
 0020 ---->         READ (JREAD,133) CNTR(M),CORR(M)
 0021 (3)         133 FORMAT (19X,F7.2,9X,F7.2)
 0022               WRITE (JPRINT,151) M,CNTR(M),CORR(M)
 0023             151 FORMAT (' ',4X,I5,9X,F7.2,9X,F7.2)
 0024             132 CONTINUE
 0025            2002 CONTINUE
            C     WT ARRAY DATA READ IN.
 0026 ---->         READ (JREAD,1012) SIG1,MWTLO1,WTC1,WTF1,SIG2,MWTLO2,WTC2,WTF2
 0027            1012 FORMAT (F5.0,I5,2F5.2,F5.0,I5,2F5.2)
            C     SIG1 IS USED AS THE SIGMA IN WT FOR SMOOTHING
            C     MWTLO1 IS USED AS THE CHANNEL NO AT WHICH THE SMOOTHING BEGINS
            C     SIG2 IS THE SIGMA USED FOR WT IN IMP RES. CASE
            C     MWTLO2 IS THE CHANNEL NO ABOUT WHICH THE IMP RES TAKES EFFECT.
            C     MWTLO2 SHOULD BE LARGER THAN 3*SIG2 TO KEEP J1 FROM BEING LEQ 0.
            C     IF MWTLO2 IS LEQ 0 NO IMP RES IS DONE.
```

```
FCRTRAN IV G LEVEL 1, MCD 2          MAIN              DATE = 68222          11/31/52

            C     WTC1 AND WTF1 ARE CONSTANTS USED IN WT SMCOTHING.
            C     WTC2 AND WTF2 ARE CONSTANTS USED IN WT IMPROVED RESOLUTION
0028         2003 CONTINUE
0029              READ (JREAD,121) J2,FIRENG,DELENG
0030         121  FORMAT(I5,2F5.0)
            C     J2       NUMBER OF POINTS IN THE EFFCY ARRAY
            C     FIRENG IS THE FIRST ENERGY (KEV) VALUE USED IN EFFICIENCY CALC.
            C     DELE IS THE ENERGY (KEV) DIFFERENCE BETWEEN EFFICIENCY POINTS.
            C     EFFCY   EFFICIENCY DATA ARRAY.
0031              READ (JREAD,120) (EFFCY(I),I=1,J2)
0032         120  FORMAT (7E10.3)
            C     DATA READ IN
0033         2004 CONTINUE
0034              READ (JREAD,1001) NUMRUN,NOCHAN,IMAX,DCR,ECR,BGER,IPUNCH,ERTM,
                1 SANGLE,FLUXT
0035         1001 FORMAT (3I5,F5.0,2F5.1,I5,F5.2,2E10.4)
            C     NUMRUN   THE DATA RUN NUMBER.
            C     NOCHAN   NUMBER OF CHANNELS IN THE DATA RUN.
            C     GENERALLY NOCHAN >  2**N - 1
            C     IMAX     CHANNEL NUMBER SLOPE CRITERION, CHANNELS LEQ IMAX ARE NOT
            C     ANALYZED.
            C     DCR      SLOPE CRITERION USED FOR BACKGROUND CALCULATION.
            C     ECR      ERROR CRITERION USED IN SUBROUTINE PKANAL.
            C     BGER     THE CORRECTION FACTOR USED IN PKLOC TO LOCATE THE PEAKS.
            C     IPUNCH LESS THAN ZERO PUNCHES OUT PEAK ANALYSIS DATA
            C     IPUNCH EQUAL ZERO IMPLIES NO PUNCHED OUTPUT.
            C     IPUNCH EQUAL 1 PUNCHES OUT THE SMOOTHED DATA
            C     IPUNCH EQUAL 2 PUNCHES OUT THE SMOOTHED, BACKGROUND SUBTRACTED DATA
            C     IPUNCH EQUAL 3 PUNCHES OUT THE IMPROVED RESOLUTION DATA.
            C     ERTM IS THE ERROR TERM CORRECTION FACTOR DUE TO SMOOTHING.  1.69
            C     SANGLE IS THE SOLID ANGLE CORRECTION TERM
            C     FLUXT IS THE NUMBER OF CAPTURES/100 IN THE SAMPLE.
0036              READ (JREAD,100) (TBK(I),I=1,NOCHAN)
            C     TBK IS THE DATA ARRAY
0037              NUM = NOCHAN + 1
            C     NUM SHOULD BE A POWER OF TWO FOR FASTEST RESULTS.
0038              TBK(NUM) = 0.0
0039              SUMOR = 0.0
0040              DO 1520 I = IMAX,NOCHAN
            C     SUMOR IS THE SUM OF NOCHAN-IMAX CRIGINAL DATA POINTS.
0041         1520 SUMOR = SUMOR + TBK(I)
0042              WRITE (JPRINT,1014) NUMRUN
0043         1014 FORMAT (1H1,4X,25H ORIGINAL DATA OF RUN NO ,I5)
0044              LK = NOCHAN/10
0045              DO 1501 J = 1,LK
0046              JU = J*10
0047              JL = JU - 9
            C     JU IS THE CHANNEL NUMBER FOR EVERY TENTH POINT
```

```
0048                  WRITE (JPRINT,1500) (TBK(I),I=JL,JU),JU
0049                1500 FORMAT(2X,10(F9.2,2X),I6)
0050                1501 CONTINUE
0051                  JA = JU+1
0052                  WRITE (JPRINT,1502) (TBK(I),I=JA,NOCHAN)
0053                1502 FORMAT(2X,10(F9.2,2X))
                    C
                    C     START OF SMOOTHING OF TBK
                    C     CALCULATION OF THE WEIGHTING ARRAY WT.
0054                  DO 134 I=1,NUM
0055                 134 WT(I) = WTC1 + WTF1
0056                  SIGSQ = SIG1*SIG1
0057                  NUP = NUM/2
0058                  DO 135 I=MWTLO1,NUP
0059                  WTPO = (FLOAT(I - MWTLC1) * FLOAT(I - MWTLO1))/SIGSQ
0060                  WT(I) = WTC1 + WTF1 * EXP(-WTPO/2.0)
0061                  J = NUM - I
0062                  WT(J) = WT(I)
0063                 135 CONTINUE
                    C      SUBROUTINE CTFFT PERFORMS THE COOLEY-TUKEY FAST FOURIER TRANSFORM
                    C     CTFFT CALCULATES THE FOURIER TRANSFORM, FILTERS THE TRANSFORMED
                    C     DATA WITH WT AND PERFORMS THE INVERSE TRANSFORM.
                    C     THE INPUT ARRAY TBK IS REPLACED BY THE FILTERED DATA AT OUTPUT.
                    C
0064                        CALL CTFFT (DATA,TBK,WT,NUM,JPRINT)
                    C
0065                  WRITE (JPRINT,1015) NUMRUN,SIG1,MWTLC1,WTC1,WTF1
0066                1015 FORMAT ('1',4X,' SMOOTHED DATA OF RUN NO ',I5,' SIG1 = ',F6.0,
                       1 ' MWTLO1 = ',I5,' WTC1 = ',F5.2,' WTF1 = ',F5.2)
0067                  DO 1503 J = 1,LK
0068                  JU = J*10
0069                  JL = JL - 9
0070                  WRITE (JPRINT,1504) (TBK(I),I=JL,JU),JU
0071                1504 FORMAT(2X,10(F9.2,2X),I6)
0072                1503 CONTINUE
0073                  JA = JU+1
0074                  WRITE (JPRINT,1502)(TBK(I),I=JA,NOCHAN)
0075                  IF (IPUNCH-1) 1523,1521,1523
0076                1521 WRITE (JPUNCH,100) (TBK(I),I=1,NOCHAN)
0077                1523 CONTINUE
0078                  SUMSM = 0.0
0079                  DO 157 I=IMAX,NOCHAN
0080                 157 SUMSM = SUMSM + TBK(I)
                    C
                    C     BACKGROUND SUBTRACTION
0081                        CALL BAKSUB (NOCHAN,IMAX,DCR)
                    C     SUBROUTINE BACKSUB PERFORMS THE BACKGROUND SUBTRACTION ON ARRAY
                    C     TBK.  THE ORIGINAL TBK ARRAY IS REPLACED BY THE BACKGROUND
```

```
                C       SUBTRACTED ARRAY.
                C       SUBRCUTINE BAKSUB USES TBL ARRAY TC STORE THE BACKGROUND.
 CC82                   WRITE (JPRINT,154) NUMRUN
 CC83             154   FORMAT ('1',' CALCULATED BACKGRCUND CF RUN NO ',I5)
 CC84                   LLO = 1
 CC85                   LUP = 10
 CC86             155   CONTINUE
 CC87                   WRITE (JPRINT,1504) (TBL(I),I=LLC,LUP),LUP
 CC88                   LLO = LLC + 10
 CC89                   LUP = LUP + 10
 CC90                   IF (LUP - NOCHAN) 155,156,156
 CC91             156   CONTINUE
 CC92                   WRITE (JPRINT,1502) (TBL(I),I=LLC,NCCHAN)
 CC93                   WRITE (JPRINT,1016) NUMRUN
 CC94            1016   FORMAT ('1',4X,'SMCOTHED BACKGRCUND SUBTRACTED DATA RUN NO ',I5)
 CC95                   DO 1506 J = 1,LK
 CC96                   JU = J*10
 CC97                   JL = JU - 9
 CC98                   WRITE (JPRINT,1507) (TBK(I),I=JL,JU),JU
 CC99            1507   FORMAT(2X,10(F9.2,2X),I6)
 C100            1506   CONTINUE
 C101                   JA = JU+1
 01C2                   WRITE (JPRINT,1502) (TBK(I),I=JA,NCCHAN)
 C103                   SUMBG = 0.0
 C104                   SUMBK = 0.0
 C105                   DO 153 I = IMAX,NOCHAN
                C       SUMBG IS THE SUM CF NOCHAN-IMAX SMCOTHED BACKGROUND SUBTRACTED
                C       DATA PCINTS.
 01C6                   SUMBG = SUMBG + TBK(I)
                C       SUMBK IS THE SUM CF NOCHAN-IMAX BACKGROUND DATA POINTS.
 01C7                   SUMBK = SUMBK + TBL(I)
 C108             153   CONTINUE
 C109                   IF (IPUNCH-2) 1525,1524,1525
 0110            1524   WRITE (JPUNCH,100) (TBK(I),I=1,NCCHAN)
 C111            1525   CONTINUE
 0112                   WRITE (JPRINT,550)
 0113             550   FORMAT ('1',4X,' SUMATICNS FRCM IMAX ')
 0114                   WRITE (JPRINT,551) SUMCR
 0115             551   FORMAT(1HO,4X, 30H TCTAL CCUNTS IN ORIGINAL DATA, F10.0)
 0116                   WRITE (JPRINT,552) SUMSM
 0117             552   FORMAT(1HO,4X, 30H TOTAL CCUNTS IN SMCOTHED DATA, F10.0)
 0118                   WRITE (JPRINT,553) SUMBK
 0119             553   FORMAT(1HO,4X, 30H TOTAL BACKGRCUND CCUNTS      , F10.0)
 0120                   WRITE (JPRINT,554) SUMBG
 C121             554   FORMAT ('0',4X,' TOTAL CCUNTS IN SM-BKSUB DATA',F10.0)
                C
                C       IMPRCVED RESOLUTICN SECTICN
                C       IF MWTLC2 IS LEQ 0 NO IMP RES IS DCNE.
```

```
0122                IF (MWTLO2) 142,142,143
0123              143 CONTINUE
0124                DO 136 I=1,NUM
0125                WT(I) = WTO2
0126              136 CONTINUE
                C     CALCULATION OF WT FOR IMPROVED RESOLUTION CASE.
0127                SIGSQ = SIG2*SIG2
0128                MSIG = SIG2
0129                NUP = MWTLO2 + 3 * MSIG
                C     NOTE LIMITS ON NUP OF 3 * SIG2.
0130                DO 137 I = MWTLO2,NUP
0131                WTPO = FLOAT(I-MWTLO2)*FLOAT(I-MWTLO2)/SIGSQ
0132                WT(I) = WTO2 + WTF2 * EXP(-WTPO/2.0)
0133                J1 = MWTLO2 - I + MWTLO2
0134                WT(J1) = WT(I)
0135                J4 = NUM - MWTLO2 + I - MWTLO2
0136                WT(J4) = WT(I)
0137                J3 = NUM - I
0138                WT(J3) = WT(I)
0139              137 CONTINUE
                C
0140                      CALL CTFFT (DATA,TBK,WT,NUM,JPRINT)
                C
                C     NEGLECTING ALL CHANNELS WITH COUNTS LEQ 20.0.
0141                DO 138 I = 1,NUM
0142                IF (TBL(I)) 162,162,163
0143              162 CONTINUE
0144                TBL(I) = 0.0
0145              163 CONTINUE
0146                DUM = TBK(I) - 20.0
0147                IF (DUM) 139,139,138
0148              139 TBK(I) = 0.0
0149              138 CONTINUE
0150                WRITE (JPRINT,140) NUMRUN
0151              140 FORMAT ('1',4X,' SMOOTHED , BACKGROUND SUBTRACTED, IMPROVED RESOLU
                   1TION DATA OF RUN NO ',I5)
0152                WRITE (JPRINT,150) SIG2,MWTLO2,WTO2,WTF2
0153              150 FORMAT (' ',4X,' SIG2 = ',F5.0,5X,' MWTLO2 = ',I5,5X,' WTO2 = ',
                   1 F5.2,5X,'WTF2 = ',F5.2)
0154                DO 141 J=1,LK
0155                JU = J * 10
0156                JL = JU - 9
0157                WRITE (JPRINT,1507) (TBK(I),I=JL,JU),JU
0158              141 CONTINUE
0159                JA = JU + 1
0160                WRITE (JPRINT,1502) (TBK(I),I=JA,NOCHAN)
0161                IF (IPUNCH-3) 142,144,142
0162              144 WRITE (JPUNCH,100) (TBK(I),I=1,NOCHAN)
```

```
0163          142 CONTINUE
              C
0164              WRITE (JPRINT,159)
0165          159 FORMAT ('1',' BACKGROUND DATA FOR EVERY TENTH POINT ')
0166              J10 = NOCHAN/10
0167              DO 160 I =1,J10
0168              K = 10 * I - 9
0169          160 TBL(I) = TBL(K)
0170              WRITE (JPRINT,161) (TBL(I),I=1,J10)
0171          161 FORMAT ((' ',10(F9.2,2X)))
              C       CALIBRATION OF DATA USING TWO ENERGY STANDARDS.
0172              READ (JREAD,1002) EGAM1,IP1,EGAM2,IP2,FPS,ERFW
0173         1002 FORMAT (4X,F6.1,4X,I6,4X,F6.1,4X,I6,2X,F7.2,3X,F7.2)
              C       EGAM1 IS THE ENERGY (KEV) OF THE FIRST CALIB.PEAK.
              C       EGAM1 LEQ ZERO DELETES ANALYSIS OF THE GAMMA PEAKS.
              C       IP1 IS THE PEAK CENTER OF THE FIRST PEAK.
              C       EGAM2 IS THE ENERGY (KEV) OF THE SECOND CALIB. PEAK.
              C       IP2 IS THE PEAK CENTER OF THE SECOND PEAK.
              C       FPS IS THE FIRST PEAK SIGMA USED IN PKANAL TO FIT THE FWHM.
              C       FPS LEQ 0 IMPLIES FPS = FWHM OF FIRST CALIB.PEAK.
              C       ERFW IS THE RANGE THE FWHM IS ALLOWED TO DEVIATE FROM FPS.
              C       FPS AND ERFW ARE GIVEN IN KEV.
0174              IF (EGAM1) 146,146,145
0175          145 CONTINUE
0176                    CALL PKALIB (NOCHAN,EGAM1,IP1,EGAM2,IP2,SLP,WID1,WID2,
                     1 TPC1,TPC2)
              C       SUBROUTINE PKALIB CALCULATES THE FWHM, TRUE PEAK CENTERS OF THE
              C       CALIBRATION PEAKS, ANS SLP TO CONVERT FROM CHANNEL NO. TO ENERGY.
              C       PKALIB USES ADJUST TO CORRECT FOR THE NONLINEARITY OF THE SYSTEM.
0177              WRITE (JPRINT,122) NUMRUN, NOCHAN
0178          122 FORMAT (1H1,4X,9H RUN NO =,I5,5X,22H NUMBER OF CHANNELS = ,I5)
0179              WRITE (JPRINT,123) IMAX
0180          123 FORMAT (5X,41H CHANNEL NUMBER SLOPE CRITERION (IMAX) = ,I5)
0181              WRITE (JPRINT,126) ECR
0182          126 FORMAT (' ',4X,' ERROR CRITERION FOR PEAK ANALYSIS (PER-CENT) ',
                     1 F5.1)
0183              WRITE (JPRINT,125) DCR
0184          125 FORMAT(5X,51H SLOPE CRITERION FOR BACKGROUND IN UNITS OF SQRT OF,
                    114H BACKGROUND = , F6.2)
0185              WRITE (JPRINT,1020) BGER
0186         1020  FORMAT (5X,65H PEAK LOCATION CRITERION IN UNITS OF SQRT OF BACKGR
                    1OUND (BGER) = ,F5.1)
0187              WRITE (JPRINT,158) ERTM
0188          158 FORMAT (' ',4X, ' ERROR TERM REDUCTION DUE TO SMOOTHING ',F5.3)
0189              WRITE (JPRINT,1018) LR,N2,N3,XN
0190         1018  FORMAT (5X,21H LINEARITY RUN NUMBER,I5,5X,5H N2 =,I5,5X,
                     1 5H N3 =,I5,5X,5H XN =,F5.0)
0191              WRITE (JPRINT,1019) FPS,ERFW
```

```
FORTRAN IV G LEVEL 1, MOD 2            MAIN              DATE = 68222          11/31/52

 0192          1019 FORMAT (' ',4X,' FIRST PEAK FWHM (KEV) FPS = ',F7.2,5X,
                    1 ' FWHM VARIATION (KEV) ERFW = ',F7.2)
 0193               WRITE (JPRINT,130) SLP
 0194           130 FORMAT (' ',4X,' ENERGY PER CH.NO (KEV) = ',F6.3)
 0195               WRITE (JPRINT,128) EGAM1,IP1,TPC1,WID1
 0196           128 FORMAT (' ',4X,' ENERGY-1 = ',F6.1,4X,' CHANNEL NO = ',I6,
                    1 4X,' TRUE PEAK CENTER = ',F6.1,4X,' WIDTH (KEV) = ',F6.2)
 0197               WRITE (JPRINT,129) EGAM2,IP2,TPC2,WID2
 0198           129 FORMAT (' ',4X,' ENERGY-2 = ',F6.1,4X,' CHANNEL NO = ',I6,
                    1 4X,' TRUE PEAK CENTER = ',F6.1,4X,' WIDTH (KEV) = ',F6.2)
 0199               WRITE (JPRINT,149) SANGLE,FLUXT
 0200           149 FORMAT (' ',4X,' SOLID ANGLE RADIANS = ',E10.4,2X,' FLUXT = ',
                    1 E10.4)
 0201               WRITE (JPRINT,147) FIRENG,DELENG
 0202           147 FORMAT (' ',4X,' EFFICIENCY DATA INITIAL ENERGY (KEV) = ',F5.0,
                    1 2X,' DELTA ENERGY (KEV) = ',F5.0)
 0203               WRITE (JPRINT,148) (EFFCY(I),I=1,J2)
 0204           148 FORMAT ((' ',4X,10(E10.3)))
                C
                C      LOCATION AND NUMBERING OF THE PEAKS
 0205                      CALL PKLOC (NOCHAN,NI,IMAX,BOER)
                C      SUBROUTINE PKLOC LOCATES AND NUMBERS THE PEAKS OF THE TBK ARRAY
                C      USING A FIRST DIFFERENCE CALCULATION IN THE CTS ARRAY.
                C      NI       NUMBER OF THE PEAK.
                C
                C
                C      ANALYSIS OF THE GAMMA PEAKS
 0206                      CALL PKANAL (NI,SLP,ECR,TPC1,EGAM1,WID1,FPS,ERFW,ERTM,
                    1 FIRENG,DELENG,SANGLE,FLUXT)
                C      SUBROUTINE PKANAL ANALYZES THE THE GAMMA PEAKS FOR AREA AND WIDTH
                C      FROM SINGLETS TO TRIPLETS.
 0207           146 CONTINUE
                C
 0208                READ (JREAD,1008) MO
 0209           1008 FORMAT (I5)
 0210                GO TO (2001,2002,2003,2004), MO
                C      MO=1 PROGRAM LOOPS TO 2001,MO=2 PROGRAM LOOPS TO 2002, ETC.
                C      MO.LEQ.0 OR MO.GTR.4 ENDS PROGRAM BY GOING TO 2010
 0211           2010 CONTINUE
 0212                CALL EXIT
 0213                END
```

```
FORTRAN IV G LEVEL 1, MOD 2            LINEAR            DATE = 68214          20/05/11

0001                SUBROUTINE LINEAR (LR,N2,N3,XN,NOCHAN)
                C   SUBROUTINE LINEAR CALCULATES THE LINEARITY CORRECTION FACTOR
                C   USING THE LINEARITY DATA GIVEN IN ARRAY TBL.
                C   THE CORRECTION FACTOR, CORR, IS IN THE FORM OF AN ARRAY OF
                C   DIMENSION L (L LEQ 200) OF THE L LINEARITY PEAKS CHARACTERIZING
                C   THE DATA OF TBL.
                C   THE ADJUST SUBROUTINE USES THE CORR FACTOR FOR ITS INTERPOLATION
                C   TO CORRECT FOR LINEARITY.
                C   XN       IS THE LIMIT OF THE LINEARITY BACKGROUND.  ONLY TRUE
                C            CALIBRATION PEAKS ARE GTR THAN XN.
                C   TBL      IS THE LINEARITY DATA.
                C   N2       IS THE CHANNEL PEAK NO. OF THE LOW REFERENCE PEAK
                C   N3       IS THE CHANNEL PEAK NO. OF THE HIGH REFERENCE PEAK.
                C   CNTR(L)  IS THE PEAK CENTER AS CALC BY WEIGHTING THE FIRST MOMENT
                C            OF THE PEAK TO THE INTEGRAL UNDER THE PEAK.
                C   CORR(L)  IS THE LINEARITY CORRECTION FACTOR FOR THE L-TH PEAK.
                C   SUM      IS THE PEAK INTEGRAL.
                C   XMOM     IS THE FIRST MOMENT OF THE PEAK.
                C   SLOPE    IS THE LINEAR SLOPE BETWEEN THE N2 AND N3 PEAKS.
0002                COMPLEX DATA
0003                COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                   1,CORR(200),EFFCY(100)
0004                COMMON IPUNCH,JREAD,JPRINT,JPUNCH
0005                J=0
0006                L=0
0007                I = 0
0008             3  I=I+J+1
0009                IF (I-NOCHAN) 31,7,7
0010            31  J=0
0011                IF (TBL(I)-XN) 3,3,4
0012             4  J=J+1
0013                IPLJ=I+J
0014                IF (IPLJ-NOCHAN) 12,7,7
0015            12  CONTINUE
0016                IF (TBL(IPLJ)-XN) 5,5,4
0017             5  SUM=0.
0018                XMOM=0..
0019                DO 6 K2=1,J
0020                K1=I+K2-1
0021                SUM=SUM+TBL(K1)
0022                XMOM=XMOM+FLOAT (K2)*TBL(K1)
0023             6  CONTINUE
0024                L=L+1
0025                IF (L-200) 13,7,7
0026            13  CONTINUE
0027                CNTR(L) = XMOM/SUM + FLOAT (I) - 1.0
0028                IF (I-NOCHAN) 3,7,7
0029             7  SLOPE=(CNTR(N3)-CNTR(N2))/FLOAT (N3-N2)
```

```
FORTRAN IV G LEVEL 1, MOD 2            LINEAR              DATE = 68214          20/05/11

 0030                WRITE (JPRINT,9) LR,N2,N3,XN
 0031              9 FORMAT ('1','  LINEARITY RUN NUMBER ',I5,'   N2 = ',I5,'   N3 = ',
                    1 I5,'    XN = ',F5.0)
 0032                WRITE (JPRINT,10)
 0033             10 FORMAT ('0',14X,' CNTR ARRAY     CORR ARRAY ')
 0034                DO 8 M=1,L
 0035                CORR(M)=(CNTR(N2)+FLOAT (M-N2)*SLOPE)-CNTR(M)
 0036                WRITE (JPRINT,11) M,CNTR(M),CORR(M)
 0037                WRITE (JPUNCH,11) M,CNTR(M),CORR(M)
 0038             11 FORMAT (' ',4X,I5,9X,F7.2,9X,F7.2)
 0039             8  CONTINUE
 0040                RETURN
 0041                END
```

```
FORTRAN IV G LEVEL 1, MOD 2            CTFFT              DATE = 68214          20/05/11

0001                  SUBROUTINE CTFFT (DATA,TBK,WT,NUM,JPRINT)
                C     COOLEY-TUKEY FAST FOURIER TRANSFORM
0002                  COMPLEX DATA(4096)
0003                  REAL TBK(4096),WT(4096)
                C     CTFFT TRANSFORMS THE ARRAY TBK, WEIGHTS IT WITH WT, AND PER-
                C     FORMS THE INVERSE TRANSFORMATION.
                C     DATA IS A COMPLEX ARRAY USED IN SUBROUTINE FOURT.
                C     SUBROUTINE FOURT DOES THE ACTUAL FFT CALCULATION.
                C     NUM SHOULD BE A POWER OF TWO FOR FASTEST RESULTS.
0004                  DO 1 I = 1,NUM
0005                  DATA(I) = CMPLX(TBK(I),0.0)
0006                1 CONTINUE
                C
                C     FAST FOURIER TRANSFORM OF DATA.
0007                        CALL FOURT (DATA,NUM,1,+1,+1,0)
                C
0008                  DO 2 I = 1,NUM
0009                  AR = REAL(DATA(I))
0010                  AI = AIMAG(DATA(I))
0011                  TBK(I) = SQRT(AR*AR + AI*AI)
                C     TBK IS USED TO STORE THE MAGNITUDE OF EACH TRANSFORMED POINT.
0012                2 CONTINUE
0013                  W = 6.2831873/FLOAT(NUM)
0014                  NUM2 = NUM/2 + 2
                C     PRINT OUT OF TRANSFORMED DATA.
0015                  WRITE (JPRINT,3) W
0016                3 FORMAT (1H1,26H DELTA OMEGA IN RADIANS = ,F10.6)
0017                  WRITE (JPRINT,4)
0018                4 FORMAT (1H ,19H TRANSFORM INTEGRAL)
0019                  WRITE (JPRINT,5) (TBK(I),I=1,NUM2)
0020                5 FORMAT (1H ,10E11.4)
                C     FILTERING THE TRANSFORMED DATA WITH WT(I).
0021                  DO 8 I = 1,NUM
                C     SMOOTHING
0022                  DATA(I) = WT(I) * DATA(I)
0023                8 CONTINUE
                C
                C     INVERSE FOURIER TRANSFORM
0024                        CALL FOURT(DATA,NUM,1,-1,+1,0)
                C
0025                  DO 10 I = 1,NUM
0026                  AR = REAL(DATA(I))
0027                  AI = AIMAG(DATA(I))
0028                  TBK(I) = SQRT(AR*AR + AI*AI)/FLOAT(NUM)
                C     DIVIDING BY NUM IS TO CORRECTLY NORMALIZE THE OUTPUT DATA.
0029               10 CONTINUE
0030                  RETURN
0031                  END
```

```
FORTRAN IV G LEVEL 1, MOD 2            BAKSUB            DATE = 68217          21/18/01

0001                  SUBROUTINE BAKSUB (NOCHAN,IMAX,DCR)
                C     SUBROUTINE BACKSUB PERFORMS THE BACKGROUND SUBTRACTION ON ARRAY
                C     TBK.  THE ORIGINAL TBK ARRAY IS REPLACED BY THE BACKGROUND
                C     SUBTRACTED ARRAY.
0002                  COMPLEX DATA
0003                  COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                     1,CORR(200),EFFCY(100)
0004                  COMMON IPUNCH,JREAD,JPRINT,JPUNCH
                C
0005                  WRITE (JPRINT,498)
0006                498 FORMAT(1H1,35X,37H CHOSEN MINIMA AND SLOPE BETWEEN THEM)
0007                  WRITE (JPRINT,499)
0008                499 FORMAT(20X,29H   NO.    L CH.    LEFT MIN   ,
                    131H RIGHT MIN       SLOPE     BASE)
                C
0009                  JJ = NOCHAN - 1
0010                  DO 302 I = 1,JJ
0011                302 DTS(I)=TBK(I+1)-TBK(I)
                C     IMAX    LOWER LIMIT ON SLOPE CRITERION.
0012                  LA = IMAX + 1
0013                  I = IMAX + 2
0014                  IMIN = 0
0015                  IX = 0
0016                  ILOOP = 0
                C     XM NEGATIVE, DB POSITIVE DETERMINES A PEAK.
0017                333 XM=DTS(I)*DTS(I-1)
0018                  DB=DTS(I)-DTS(I-1)
0019                  IF (XM) 305,305,303
0020                305 IF (DB) 303,303,304
0021                304 CONTINUE
                C
                C     EXAMINING THE SLOPES OF ADJACENT PEAKS FOR POSSIBLE MULTIPLETS
                C     MINIMA OCCURING WITHIN 15 CHANNELS FROM THE END OF THE DATA ARRAY
                C     ARE NOT AVERAGED, NOR IS THE SLOPE CRITERION APPLIED
0022                  IF(I+15-JJ) 310,310,510
                C     WHENEVER THE ANGLE BETWEEN THE SLOPES OF TWO ADJACENT PEAKS IS
                C     LARGE THE PAIR IS CONSIDERED AS A MULTIPLET.   PROCEDURE IS
                C     REPEATED TILL THERE ARE A MAXIMUM OF FIVE  PEAKS BETWEEN THE
                C     CHOSEN MINIMA, THE SLOPE CRITERION BEING IGNORED THEREAFTER.
                C
                C     GRADIENT CALCULATION FOR BACKGROUND UNDER THE PEAK.
0023                310 GRAD=(TBK(I)-TBK(LA))/FLOAT (I-LA)
0024                  IX = IX+1
                C     THE FIRST FOUR ELEMENTS OF THE TBL ARRAY ARE USED TEMPORARILY
                C     TO STORE THE SLOPES AND THE MINIMA OF TWO ADJACENT PEAKS
0025                  TBL(IX) = GRAD
0026                  TBL(IX+2) = FLOAT (I)
0027                  IF(IX-2) 303,564,564
```

```
0028          564 CONTINUE
0029              GRAD1 = TBL(1)
0030              GRAD2 = TBL(2)
                C     M1 IS THE RHS MINIMUM OF THE PEAK UNDER CONSIDERATION
                C     AND M2 THE NEXT HIGHER MINIMUM
0031              M1 = TBL(3)
0032              M2 = TBL(4)
0033              TB1 = TBK(LA) + GRAD1*FLOAT (M2-LA)
                C     TB1 IS THE VALUE TBK(M2) WOULD HAVE IF THE ANGLE BETWEEN THE
                C     TWO SLOPES WERE ZERO.
0034              TB2 = TBK(M2)
0035              TB12 = (TB1+TB2)/2.0
0036              IF(TB12) 663,663,664
0037          663 CRIT = DCR*FLOAT (M2-M1)
0038              GO TO 665
0039          664 CRIT = DCR*FLOAT (M2-M1)*SQRT (TB12)
0040          665 CONTINUE
                C     DCR IS READ IN AS DATA
0041              IF(TB1-TB2) 565,565,666
0042          666 IF(TB1-TB2-CRIT) 565,565,566
0043          566 ILOOP = ILOOP + 1
0044              IF(ILOOP-5) 567,565,565
0045          567 I = M1
0046              IX = 0
0047              GO TO 303
0048          565 ILOOP = 0
0049              IX = 0
0050              I = M1
                C
                C     THREE-POINT AVERAGING OF THE MINIMA
                C     FROM HERE TO STATEMENT 533 THE PROGRAM AVERAGES THE MINIMA
                C     BY WEIGHING THEM EQUALLY WITH THE ORIGINAL COUNTS IN THE TWO
                C     CHANNELS CLOSEST TO THE MINIMUM (ONE ON EACH SIDE)
0051           308  LL=I
0052              LOOP = 0
                C     IF THE AVERAGING PROCESS IS REPEATED MORE THAN 15 TIMES
                C     FOR ANY ONE GIVEN MINIMUM THE AVERAGING PROCESS IS IGNORED
0053           507  LOOP = LOOP + 1
0054              IF(LOOP - 16) 525,510,510
0055           525  CONTINUE
0056              TB = (TBK(LL-1)+TBK(LL)+TBK(LL+1))/3.0
                C     NEW MINIMA ARE SOUGHT
0057              IF(TB-TBK(LL-2)) 603,603,602
0058           602  IF(TB-TBK(LL-1)) 603,603,502
0059           603  IF (TB-TBK(LL+2)) 503,503,604
0060           604  IF(TB-TBK(LL+1)) 503,503,504
0061           502  LL = LL-1
                C     IF THE PROGRAM RETURNS TO THE ORIGINAL MINIMUM THE AVERAGING
```

```
      C     PROCESS IS STOPPED.
0062          IF(LL-I) 523,503,523
0063        523 CONTINUE
0064          IF(LL - LA) 303,303,507
0065        504 LL = LL + 1
0066          IF(LL-I) 524,503,524
0067        524 CONTINUE
0068          IF(LL-I-6) 507,510,510
      C     IF NO NEW MINIMUM IS FOUND WITHIN 6 CHANNELS ABOVE THE ORIGINAL
      C     MINIMUM THE AVERAGING PROCESS IS IGNORED
0069        510 LL = I
0070          GO TO 533
0071        503 CONTINUE
0072          TBK(LL) = (TBK(LL-1)+TBK(LL)+TBK(LL+1))/3.0
      C
      C     LA       IS LOWER CHAN NO LIMIT FOR LINEAR BACKGROUND FIT.
      C     LL       IS UPPER CHAN NO LIMIT FOR LINEAR BACKGROUND FIT.
0073        533 LB=LA+1
0074          LX=LL-1
      C     LBASE IS THE NUMBER OF CHANNELS OCCUPIED BY THE PEAK (FIRST CH=0)
0075          LBASE = LL - LA
      C     QSLOP IS THE SLOPE BETWEEN THE TWO MINIMA OF A PEAK
0076          QSLOP = (TBK(LL)-TBK(LA))/FLOAT (LBASE)
      C     TBL IS A DUMMY ARRAY USED FOR CALCULATING THE BACKGROUND.
0077          TBL(LA)=TBK(LA)
0078          IF (LBASE - 8) 550,551,551
0079        551 CONTINUE
0080          IMIN = IMIN + 1
0081          WRITE (JPRINT,500) IMIN,LA,TBK(LA),TBK(LL),QSLOP,LBASE
0082        500 FORMAT(20X, I7,3X,I6,3X,F9.2,3X,F9.2,3X,F9.3,3X,I5)
0083        550 CONTINUE
      C     BACKGROUND - LINEAR FIT.
0084          DO 306 IK=LB,LX
0085         306  TBL(IK)=(TBK(LA)*FLOAT (LL-IK)+TBK(LL)*FLOAT (IK-LA))/
                 1FLOAT (LL-LA)
0086          LA=LL
0087          IF(I-LL) 520,303,303
0088         520 I = LL
0089         303  IF (I-JJ) 334,335,335
0090         334  I=I+1
0091          GO TO 333
0092         335  TBL(LA)=TBK(LA)
0093          LB=LA+1
0094          LX = JJ
0095          LL = NOCHAN
0096          DO 307 IK=LB,LX
0097         307  TBL(IK)=(TBK(LA)*FLOAT (LL-IK)+TBK(LL)*FLOAT (IK-LA))/
                1 FLOAT (LL - LA)
```

```
0098                TBL(NOCHAN) = TBK(NOCHAN)
0099                DO 404 I = 1,IMAX
0100                TBL(I) = TBK(I)
0101                TBK(I) = 0.0
0102                DTS(I) = 0.0
0103                IF (TBL(I)) 405,404,404
0104              405 TBL(I) = 0.0
0105              404 CONTINUE
0106                DO 401 I = IMAX,NOCHAN
0107                TBK(I) = TBK(I) - TBL(I)
0108                DTS(I-1) = TBK(I) - TBK(I-1)
                  C     CORRECTION FOR NEGATIVE VALUES.
0109                IF (TBL(I)) 406,407,407
0110              406 TBL(I) = 0.0
0111              407 CONTINUE
0112                IF(TBK(I)) 403,403,401
0113              403 TBK(I) = 0.0
0114              401 CONTINUE
0115                RETURN
0116                END
```

```
FORTRAN IV G LEVEL 1, MOD 2            PKALIB            DATE = 68214          20/05/11

0001                  SUBROUTINE PKALIB (NOCHAN,EGAM1,IP1,EGAM2,IP2,SLP,WID1,WID2,
                     1 TPC1,TPC2)
                C     PKALIB USES ADJUST TO CORRECT FOR THE NONLINEARITY OF THE SYSTEM.
                C     CALIBRATION PEAKS, ANS SLP TO CONVERT FROM CHANNEL NO. TO ENERGY.
                C     SUBROUTINE PKALIB CALCULATES THE FWHM, TRUE PEAK CENTERS OF THE
0002                  COMPLEX DATA
0003                  COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                     1,CORR(200),EFFCY(100)
0004                  COMMON IPUNCH,JREAD,JPRINT,JPUNCH
0005                  JJ = NCCHAN - 1
0006                  DO 622 I = 1,JJ
0007                  DTS(I) = TBK(I+1) - TBK(I)
0008           622    CONTINUE
0009                  DTS(NOCHAN) = 0.0
0010                  JP=0
0011                  IP = IP1 - 3
0012           631    SM=DTS(IP)*DTS(IP-1)
0013                  DP=DTS(IP)-DTS(IP-1)
0014                  IF (SM) 615,615,613
0015           615    IF (DP) 617,613,613
0016           613    IP=IP+1
0017                  GO TO 631
                C     PCNR     IS PEAK CENTER.
0018           617    PCNR=FLOAT (IP)+DTS(IP)/(DTS(IP-1)-DTS(IP))+ 0.5
                C     GX IS A SECOND ORDER INTERPOLATION FIT.
0019                  GX = 0.5*(TBK(IP) + (PCNR-FLOAT(IP))*(DTS(IP)) + 0.5 *
                     1 (PCNR-FLOAT(IP))*(PCNR-FLOAT(IP+1))*(DTS(IP)-DTS(IP-1)))
                C     SUBROUTINE ADJUST CALCULATES THE TRUE PEAK CENTER, TPC, USING
                C     DATA FROM THE LINEARITY SUBROUTINE.
0020                        CALL ADJUST (CNTR,CORR,PCNR,TPC)
0021                  IF (JP) 620,620,621
                C     TPC1     TRUE PEAK CENTER FOR THE LOW ENERGY STANDARD.
0022           620    TPC1=TPC
                C     GX1      HALF MAXIMUM OF LOW ENERGY STANDARD.
0023                  GX1=GX
0024                  JP=JP+1
0025                  IP = IP2 - 3
0026                  GO TO 631
                C     TPC2     TRUE PEAK CENTER FOR THE HIGH ENERGY STANDARD.
0027           621    TPC2=TPC
                C     GX2      HALF MAXIMUM OF HIGH ENERGY STANDARD.
0028                  GX2=GX
                C     SLP      SLOPE USED TO CONVERT CHANNEL NUMBER TO ENERGY.
0029                  SLP=(EGAM2-EGAM1)/(TPC2-TPC1)
0030                  JP1=0
0031                  GX=GX1
0032                  NP=IP1
0033           669    J=NP
```

```
0034          672  IF (TBK(J)-GX) 670,670,671
0035          671  J=J-1
0036               GO TO 672
              C    PHL     CHANNEL NUMBER OF LOW ENERGY SIDE FOR FWHM CALCULATION.
0037          670  PHL=FLOAT (J)+(GX-TBK(J))/(TBK(J+1)-TBK(J))
0038               J=NP
0039          682  IF (TBK(J)-GX) 680,680,681
0040          681  J=J+1
0041               GO TO 682
              C    FWHM    FULL WIDTH AT HALF MAXIMUM.
0042          680  FWHM=FLOAT (J)-(GX-TBK(J))/(TBK(J-1)-TBK(J))-PHL
0043               WIDD = FWHM * SLP
              C    WIDD IS FWHM IN KEV
0044               IF (JP1) 686,686,687
0045          686  WID1 = WIDD
              C    WID1 IS FWHM (KEV) OF FIRST CALIB PEAK.
              C    THE VARIANCE OF THE PEAK CAN BE OBTAINED FROM THE FWHM.
              C    JP1 = 0 AT BEGINNING.
0046               GX=GX2
0047               JP1=JP1+1
0048               NP=IP2
0049               GO TO 669
0050          687  WID2 = WIDD
              C    WID2 IS FWHM (KEV) OF SECOND CALIB PEAK.
0051               RETURN
0052               END
```

```
FORTRAN IV G LEVEL 1, MOD 2            ADJUST            DATE = 68214          20/05/11

0001                  SUBROUTINE ADJUST (CNTR,CORR,PCNR,TPC)
0002                  REAL CNTR(200),CORR(200)
                C     SUBROUTINE ADJUST FINDS THE TRUE PEAK CENTER, TPC, BY CORRECTING
                C     THE DATA PEAK CENTER, PCNR, BY GX.
                C     GX IS DETERMINED BY A SECOND ORDER INTERPOLATION USING CNTR, THE
                C     CALCULATED PEAK CENTER, AND CORR, THE LINEARITY CORRECTION FACTOR
                C     FROM THE LINEARITY SUBROUTINE.
                C     A CALL TO LINEAR MUST PRECEDE A CALL TO ADJUST.
0003                  IOTA=1
0004                11 IF (CNTR(IOTA)-PCNR) 10,10,20
0005                10 IOTA=IOTA+1
0006                   GO TO 11
0007                20 G1=CORR(IOTA-1)
0008                   G2=CORR(IOTA)
0009                   G3=CORR(IOTA+1)
0010                   C1=CNTR(IOTA-1)
0011                   C2=CNTR(IOTA)
0012                   C3=CNTR(IOTA+1)
0013                   CX=PCNR
0014                   D12=(G2-G1)/(C2-C1)
0015                   D23=(G3-G2)/(C3-C2)
0016                   D123=(D23-D12)/(C3-C1)
0017                   GX = G1 + (CX-C1)*D12 + ((CX-C1)*(CX-C2)*D123)
0018                   TPC=PCNR+GX
0019                   RETURN
0020                   END
```

```
FORTRAN IV G LEVEL 1, MOD 2            PKLOC              DATE = 68214          20/05/11

0001                  SUBROUTINE PKLOC (NOCHAN,NI,IMAX,BGER)
                C     SUBROUTINE PKLOC LOCATES AND NUMBERS THE PEAKS OF THE TBK ARRAY
                C     USING A FIRST DIFFERENCE CALCULATION IN THE DTS ARRAY.
                C     NI       NUMBER OF THE PEAK.
0002                  COMPLEX DATA
0003                  COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                     1,CORR(200),EFFCY(100)
0004                  COMMON IPUNCH,JREAD,JPRINT,JPUNCH
0005                  NI = 0
0006                  JJ = NOCHAN - 1
0007                  I = IMAX
                C     ZM LEQ ZERO, DS LT ZERO DETERMINES A PEAK MAXIMUM.
0008                731 ZM=DTS(I)*DTS(I-1)
0009                    DS=DTS(I)-DTS(I-1)
0010                    IF (ZM) 715,715,713
0011                715 IF (DS) 717,713,713
0012                717 CONTINUE
                C     PEAKS WITH COUNTS LEQ CTLIM AT THEIR MAXIMUM WON'T BE ANALYZED.
0013                    K = I/10 + 1
0014                    IF(TBL(K)) 719, 720, 720
0015                719 CTLIM = 10.0
0016                    GO TO 721
0017                720 CONTINUE
0018                    CTLIM = BGER*SQRT(TBL(K)) + 10.0
0019                721 CONTINUE
0020                    IF (TBK(I)-CTLIM) 713,713,718
0021                718 CONTINUE
0022                    NI=NI+1
                C     NI       NUMBER OF THE PEAK.
                C     THE TBL DATA IS DESTROYED AND PTS 1 TO 1000 ARE USED FOR THE
                C     BACKGROUND DATA ARRAY.  PTS 1001 AND UP ARE USED FOR LOCATING THE
                C     GAMMA PEAKS BY CHANNEL NUMBER FOR PEAK MAXIMUM.
0023                    TBL(NI+1000)=I
                C     TBL(NI+1000) IS THE CHANNEL NUMBER OF THE PEAK
0024                    IF (NI-595) 713,713,732
0025                713 CONTINUE
0026                    I = I+1
0027                    IF (I-JJ) 731,731,732
0028                732 CONTINUE
0029                    DO 733 J=1,5
0030                    NU = NI + J + 1000
                C     TBL(NU) DUMMY VARIABLE FOR PKANAL SUBROUTINE.
0031                    TBL(NU) = TBL(NI+1000) + 15.0
0032                733 CONTINUE
0033                    RETURN
0034                    END
```

```
FORTRAN IV G LEVEL 1, MOD 2          INTSTY          DATE = 68214          20/05/11

0001                  SUBROUTINE INTSTY (SAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
                C     INTSTY FINDS THE EFFICIENCY OF THE SYSTEM AT THE GAMMA PEAK
                C     ENERGY , EGAM, BY A SECOND ORDER INTERPOLATION USING THE EFFICIENCY
                C     DATA IN THE EFFCY ARRAY AS THE KNOWN POINTS.
                C     SAREA   TOTAL COUNTS PER GAMMA PEAK.
                C     FIRENG IS THE FIRST ENERGY (KEV) VALUE USED IN EFFICIENCY CALC.
                C     DELE IS THE ENERGY (KEV) DIFFERENCE BETWEEN EFFICIENCY POINTS.
                C     SANGLE IS THE SOLID ANGLE CORRECTION TERM
                C     FLUXT IS THE NUMBER OF CAPTURES/100 IN THE SAMPLE.
                C     AIGAM IS THE PEAK AREA CORRECTED FOR EFFCY. SOLID ANGLE, AND FLUX.
0002                  COMPLEX DATA
0003                  COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                     1,CORR(200),EFFCY(100)
0004                  COMMON IPUNCH,JREAD,JPRINT,JPUNCH
0005                  IEGAM=2
0006                8 XEGAM = DELENG*FLOAT(IEGAM-1) + FIRENG
0007                  IF (XEGAM-EGAM) 26,27,27
0008               26 IEGAM=IEGAM+1
0009                  GO TO 8
0010               27 E1 = XEGAM - DELENG
0011                  E2=XEGAM
0012                  E3 = XEGAM + DELENG
0013                  G1=EFFCY(IEGAM-1)
0014                  G2=EFFCY(IEGAM)
0015                  G3=EFFCY(IEGAM+1)
0016                  D12=(G2-G1)/(E2-E1)
0017                  D23=(G3-G2)/(E3-E2)
0018                  D123=(D23-D12)/(E3-E1)
                C     GX       SECOND ORDER INTERPOLATED EFFICIENCY AT GAMMA ENERGY EGAM.
0019                  GX=G1+(EGAM-E1)*D12+(EGAM-E2)*(EGAM-E1)*D123
0020                  AIGAM = SAREA/(GX*SANGLE*FLUXT)
0021                  RETURN
0022                  END
```

```
FORTRAN IV G LEVEL 1, MOD 2            PKANAL          DATE = 68223        19/29/33

0001                SUBROUTINE PKANAL (NI,SLP,ECR,TPC1,EGAM1,WID1,FPS,ERFW,ERTM,
                   1 FIRENG,DELENG,SANGLE,FLUXT)
                C   SUBROUTINE PKANAL
                C   SUBROUTINE PKANAL ANALYZES THE GAMMA PEAKS FOR AREA, WIDTH,
                C   HEIGHT, MINIMA, BASE WIDTH AND BASE SLOPE
0002                COMPLEX DATA
0003                COMMON DATA(4096),TBL(4096),TBK(4096),WT(4096),DTS(4096),CNTR(200)
                   1,CORR(200),EFFCY(100)
0004                COMMON IPUNCH,JREAD,JPRINT,JPUNCH
0005                NUMPK = NI
0006                WRITE (JPRINT,900) NUMPK
0007             900 FORMAT (5X,19H NUMBER OF PEAKS = ,I5)
0008                WRITE (JPRINT,1001)
0009            1001 FORMAT(1HO,5X,48HAREA(A) = SUM(COUNTS), AREA(B) = FITTED GAUSSIAN)
                C
                C   EVALUATION OF LEAST-SQUARE-FITTED EQUATION FOR FWHM
                C   ONLY STRONG PEAKS (INCLUDING WELL RESOLVED MULTIPLETS) OF
                C   REASONABLE FWHM ARE CONSIDERED - 3 IF-STATEMENTS DETERMINE THAT
                C
                C   PEAK PARAMETERS CALCULATED HERE ARE STORED IN THE TBL ARRAY
                C   AND USED AGAIN LATER
0010                WRITE (JPRINT,1850)
0011            1850 FORMAT(1H1,25X, 36H PEAKS USED IN COMPUTING LSF OF FWHM)
0012                WRITE (JPRINT,1851)
0013            1851 FORMAT(1HO,94H   NUMBER   PEAK NO.    ENERGY KEV       WIDTH KEV
                   1      WEIGHT      PK HEIGHT   AREA/GAUSSIAN)
0014                WEITT = 0.0
0015                WE0 = 0.0
0016                WE1 = 0.0
0017                WE2 = 0.0
0018                E1 = 0.0
0019                E2 = 0.0
0020                E3 = 0.0
0021                E4 = 0.0
0022                AREAF = 0.0
0023                AREAM = 0.0
                C   FPS EXPECTED FWHM (KEV) OF FIRST PEAK EITHER READ IN OR CALC.
0024                IF (FPS) 3034,3034,3035
0025             3034 FPS = WID1
                C   WID1 IS FWHM (KEV) OF FIRST CALIB. PEAK.  CALC. IN PKALIB.
0026             3035 CONTINUE
0027                FC = 6.0
                C   FC SETS THE VALUE OF THE FIRST CONDITION ON THE PEAK HEIGHT
0028                IX = 0
0029                MM = 0
0030                I = 1
0031                TBL (1601) = 0.0
                C   FIRST PEAK NOT CONSIDERED HERE ( SEE STATEMENT 4109 FOR REASON).
```

```
                C
                C     LOOP FOR LSF OF FWHM EQN.  END OF LOOP AT 1818
0032             1800 I = I + 1
0033                  IF(I - NUMPK) 1817,1817,1818
0034             1817 CONTINUE
                C     THE FOLLOWING STATEMENT IS TO BE USED LATER TO IDENTIFY THE
                C     PEAKS WHOSE PARAMETERS HAVE BEEN CALCULATED IN EVALUATING THE
                C     LSF FWHM
0035                  TBL(I + 1600) = 0.0
                C
                C     COMPARISON OF THE PEAK HEIGHT TO THE BACKGROUND
                C     L1 IS THE CHANNEL NUMBER OF THE HIGHEST POINT IN THE PEAK
0036                  L1 = TBL(I + 1000)
0037                  J = L1
0038                  K = J/10 + 1
                C     ERROR TERM CRITERIA.  ECR IS READ IN .
                C     PEAKS MUST HAVE PREDICTED ERROR IN AREA LESS THAN ECR.
                C     ER IS PER CENT ERROR IN AREA.  TBK(J) IS PKHT. TBL(K) IS BKGND.
0039                  ER = 100.0 * SQRT(2.0 * (TBK(J) + 1.5 * TBL(K)))/(ERTM*TBK(J))
0040                  IF (ER-ECR) 1807,1807,1800
0041             1807 CONTINUE
0042                  ASB = TBL(K) + 80.
                C     THE TERM 80.0 SETS THE MINIMUM VALUE OF ACCEPTABLE PEAK HEIGHT
                C     TO ABOUT 9.0*FC(MIN) = 27.
0043                  SQ = SQRT (ASB)
                C     FIRST CONDITION   -  PEAK HEIGHT FAIRLY LARGE
0044                  IF(TBK(J) - FC*SQ) 1800,1800,1801
0045             1801 CONTINUE
                C     IF MORE THAN EIGHT CONSECUTIVE PEAKS DO NOT MEET THE ABOVE
                C     REQUIREMENTS THEN FC IS REDUCED SO AS TO MAKE SURE THAT THERE ARE
                C     NO LARGE ENERGY GAPS BETWEEN THE CHOSEN PEAKS. THE OTHER
                C     CONDITIONS STILL APPLY, HOWEVER. THE ORIGINAL VALUE OF FC IS
                C     RESTORED .
0046                  IF(I-IX-8) 1835,1835,1836
0047             1836 IF(FC-2.0) 1835,1835,1837
0048             1837 I = IX
0049                  FC = FC - 1.0
0050                  GO TO 1800
0051             1835 FC = 6.0
0052                  IX = I
                C     IA AND IL ARE THE LEFT AND RIGHT END POINTS OF THE PEAK
0053                  J = L1
0054             4801 IF(TBK(J)) 1802,1802,1803
0055             1803 J = J - 1
0056                  GO TO 4801
0057             1802 IA = J
0058                  J = TBL(I + 1000) + 1.0
0059             1804 IF(TBK(J)) 1805,1805,1806
```

```
0060          1806 J = J + 1
0061               GO TO 1804
0062          1805 IL = J
                   C
                   C      FROM HERE TO STATEMENT 4113 THE MULTIPLETS ARE IDENTIFIED.
                   C      IF THE PEAKS ARE WELL RESOLVED THE WIDTHS ARE ACCEPTED FOR
                   C      THE LEAST-SQUARES FIT. HOWEVER MULTIPLETS ARE NOT CONSIDERED
                   C      IN THE EVALUATION OF THE CORRECTION FACTOR TERA (SEE 4881)
0063          4109 LO = IBL(I+999)
0064               L2 = IBL(I + 1001)
                   C      LO AND L2 ARE THE CENTERS OF THE TWO PEAKS NEXT TO THE ONE
                   C      EXAMINED
0065               XL01 = FLOAT (L1-L0)/2.0
0066               XL12 = FLOAT (L2-L1)/2.0
                   C      BAD IS USED TO MARK THE MULTIPLETS FOR LATER IDENTIFICATION
0067               BAD = 0.0
0068               IF(LO-IA) 4110,4110,4111
0069          4111 BAD = 1.0
0070               VRNC = (FPS/(2.355*SLP))**2
                   C      VRNC IS AN APPROXIMATE VALUE OF THE VARIANCE FOR THIS CALCULATION
0071               C = EXP (-(XL01**2)/(2.0*VRNC))
                   C      IF THE CONTRIBUTION FROM A NEARBY PEAK TO THE HALF-HEIGHT
                   C      OF THE PEAK UNDER CONSIDERATION IS MORE THAN 10 PERCENT
                   C      THE SAID PEAK IS EXCLUDED FROM THE LSF.
                   C      SECOND CONDITION - PART ONE
0072               IF(IBK(LO)*C/(0.5*IBK(L1))-0.10) 4110,4110,1800
0073          4110 IF(L2-IL) 4112,4112,4113
0074          4112 BAD = 1.0
0075               VRNC = (FPS/(2.355*SLP))**2
0076               C = EXP (-(XL12**2)/(2.0*VRNC))
                   C      SECOND CONDITION - PART TWO
0077               IF(IBK(L2)*C/(0.5*IBK(L1)) - 0.10) 4113,4113,1800
0078          4113 CONTINUE
                   C
                   C      EVALUATION OF AREA UNDER THE PEAK
0079               AREA = 0.0
0080               DO 1814 K = IA,IL
0081          1814 AREA = AREA + IBK(K)
                   C      PCNR   IS CALCULATED PEAK CENTER
0082               PCNR = FLOAT (L1) + DTS(L1)/(DTS(L1-1) - DTS(L1)) + 0.5
0083                    CALL ADJUST (CNTR,CORR,PCNR,TPC)
0084               EGAM = EGAM1 + SLP*(TPC - TPC1)
0085               LY = L1
0086               PKHT = IBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                  1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
                   C      EVALUATION OF THE FWHM
0087               GX = 0.5*PKHT
0088               J=L1
```

```
0089          1842 IF(TBK(J) - GX) 1840,1840,1841
0090          1841 J = J - 1
0091               GO TO 1842
                C      PHL    CHANNEL NUMBER LOW ENERGY SIDE
0092          1840 PHL=FLOAT (J)+(GX-TBK(J))/(TBK(J+1)-TBK(J))
0093               J=L1
0094          1848 IF(TBK(J) - GX) 1845,1845,1846
0095          1846 J = J + 1
0096               GO TO 1848
0097          1845 W=FLOAT (J)-(GX-TBK(J))/(TBK(J-1)-TBK(J))-PHL
                C      W       FWHM IN CHANNEL NUMBERS.
                C      WIDTH   FWHM IN ENERGY
0098                 WIDTH=W*SLP
                C      THIRD CONDITION - PEAK MUST HAVE AN ACCEPTABLE FWHM
0099                 AFS = ABS (WIDTH - FPS)
                C      ERFW IS THE ALLOWABLE DEVIATION OF THE FWHM (KEV).
0100                 IF (AFS - ERFW) 1812,1812,1800
0101          1812 CONTINUE
                C      A NEW VALUE IS ASSIGNED TO FPS TO BE USED WITH THE NEXT PEAK
0102                 FPS = (FPS + WIDTH)/2.0
                C
                C      EVALUATION OF THE WEIGHING FUNCTION USED IN THE LEAST-SQUARES FIT
                C      WEIT IS THE WEIGHING FUNCTION USED IN THE LSF. IT IS EQUAL
                C      TO THE SQUARE OF THE INVERSE OF THE STANDARD DEVIATION IN WIDTH
                C      NOTE THAT (S.D.(FWHM)/FWHM) = +OR- (S.D.(PKHT)/PKHT) AND THAT
                C      THE S.D. IN PKHT IS SQRT(PKHT+1.5*BACKGROUND)/ERTM  WHERE
                C      THE ERTM REPRESENTS THE REDUCTION IN THE STANDARD DEVIATION
                C      INTRODUCED BY THE SMOOTHING.
0103                 SDWID = WIDTH * ER /(141.4214)
0104                 WEIT = 1.0/(SDWID*SDWID)
                C
                C      ENTERING THE PEAK PARAMETERS CALCULATED HERE IN THE TBL ARRAY
0105                 TBL(I + 420) = PCNR
0106                 TBL(I +1600) = EGAM
0107                 TBL(I+2200) = GX
0108                 TBL(I + 2800) = WIDTH
0109                 TBL(I+3400) = AREA
0110                 TBL(I+3750) = ER
                C
                C      COMPUTATIONS ON AREAS
0111                 AREAG = 2.1289*W*GX
0112                 ERA = AREA/AREAG
                C      ERA IS THE RATIO OF THE AREA UNDER THE PEAK TO THE AREA UNDER
                C      THE EQUIVALENT GAUSSIAN
                C      THE FOLLOWING STATEMENT EXCLUDES MULTIPLETS IN THE AREA SUMS
0113                 IF(BAD) 4880,4880,4882
0114          4880 CONTINUE
                C      ALSO EXCLUDED ARE SINGLETS WHICH ARE MORE THAN 30 PERCENT OFF.
```

```
FCRTRAN IV G LEVEL 1, MCD 2          PKANAL          DATE = 68223          19/29/33

0115                IF (ABS(ERA-1.0) - 0.30) 4881,4881,4882
0116             4881 AREAM = AREAM + AREAG
0117                  AREAH = AREAH + AREA
                 C    THE RATIO (AREAH/AREAM) WILL BE USED LATER TO EVALUATE CORRECTION
                 C         FOR THE NON-GAUSSIAN FORM OF THE PEAKS
0118             4882 CONTINUE
0119                  MM = MM + 1
0120                  WRITE (JPRINT,1813) MM,I,EGAM,WIDTH,WEIT,PKHT,ERA
0121             1813 FORMAT(1H ,4X,I5,5X,I5,5X,F10.2,5X,F10.3,5X,F10.3,5X,F10.2,5X,
                    1 F10.4)
                 C    FOLLOWING COMPUTATIONS TO BE USED IN THE EVALUATION OF THE
                 C    LEAST-SQUARE-FIT COEFFICIENTS (STATEMENT 1831)
0122                  EGM = EGAM/1000.
0123                  WID = WIDTH*WIDTH
0124                  WE0 = WE0 + WEIT*WID
0125                  WE1 = WE1 + WEIT*WID*EGM
0126                  WE2 = WE2 + WEIT*WID*EGM**2
0127                  E1 = E1 + WEIT*EGM
0128                  E2 = E2 + WEIT*EGM**2
0129                  E3 = E3 + WEIT*EGM**3
0130                  E4 = E4 + WEIT*EGM**4
0131                  WEITT = WEITT + WEIT
0132                  GO TO 1800
                 C    END OF LOOP FOR FITTING FWHM.
                 C    EVALUATION OF THE LSF COEFFICIENTS
0133             1818 CONTINUE
                 C    ALL THE PARAMETERS ARE DIVIDED BY 100.0 TO AVOID DEALING WITH
                 C    VERY LARGE NUMBERS
0134             1831 RM = WEITT/100.0
0135                  WE0 = WE0/100.0
0136                  WE1 = WE1/100.0
0137                  WE2 = WE2/100.0
0138                  E1 = E1/100.0
0139                  E2 = E2/100.0
0140                  E3 = E3/100.0
0141                  E4 = E4/100.0
0142                  DLT = RM*E2*E4 +2.0*E1*E2*E3 - E2**3 - RM*E3**2 - E4*E1**2
0143                  A0 = (WE0*E2*E4 + WE2*E1*E3 + WE1*E2*E3 - WE2*E2**2
                    1 - WE0*E3**2 - WE1*E1*E4)/DLT
0144                  A1= (WE1*RM*E4 + WE0*E2*E3 + WE2*E1*E2 - WE1*E2**2
                    1 - WE2*E3*RM - WE0*E1*E4)/DLT
0145                  A2 = (WE2*RM*E2 + WE1*E1*E2 + WE0*E1*E3 - WE0*E2**2
                    1 - WE1*RM*E3 - WE2*E1**2)/DLT
0146                  TERA = AREAH/AREAM
                 C
                 C    EVALUATION OF THE RMS ERROR IN FITTING THE DATA
0147                  QWAK = 0.0
0148                  I = 0
```

```
C149        2903 I = I+1
C150             IF(I-NUMPK) 2901,2901,2902
C151        2901 IF (TBL(I+1600)) 2903,2903,2904
C152        2904 EGAM = TBL(I+1600)
C153             WIDTH = TBL(I+2800)
C154             ER = TBL(I+3750)
C155             WEIT = 20000.0/(WIDTH*WIDTH*ER*ER)
C156             EGM = EGAM/1000.
C157             FWH2 = A0 + A1*EGM + A2*EGM*EGM
C158             QWIK = WIDTH*WIDTH - FWH2
C159             QWAK = QWAK + WEIT*QWIK*QWIK
C160             GO TO 2903
C161        2902 CONTINUE
C162             EMS = SQRT (QWAK/WEITT)
C            C   EMS IS THE RMS ERROR BETWEEN THE DATA AND THE CALCULATED APPROX.
C163             WRITE (JPRINT,1820)
C164        1820 FORMAT(1H1,20X, 37H EQUATION OF LEAST SQUARE FITTED FWHM)
C165             WRITE (JPRINT,1821) A0,A1,A2
C166        1821 FORMAT ('0',10X,' FWHM**2 =',E9.3,'  +  ',E9.3,
                1' *E(MEV) + ',F9.3,' *E(MEV)**2 ')
C167             WRITE (JPRINT,1819) MM
C168        1819 FORMAT(1H0, 10X, 21H NUMBER OF PEAKS USED, I6)
C169             WRITE (JPRINT,2905) EMS
C170        2905 FORMAT(1H0, 10X, 17H RMS ERROR IN LSF, F10.4)
C171             WRITE (JPRINT,1822) TERA
C172        1822 FORMAT(1H0, 10X, 24H (AREA/GAUSSIAN AREA) = , F10.4)
            C
            C    PEAK ANALYSIS  -  PEAKS THAT HAVE ALREADY BEEN ANALYSED ABOVE
            C    HAVE THEIR PARAMETERS STORED IN THE TBL ARRAY
            C
            C    ALL TYPES OF PEAKS ARE CALCULATED IN TWO DIFFERENT WAYS
            C      (A) AREA = SUM OF COUNTS IN ALL CHANNELS UNDER PEAK (SAREA)
            C      (B) AREA = 1.0645*PEAK HEIGHT*FWHM*TERA WHERE THE TERM TERA
            C          IS A CORRECTION FACTOR FOR THE NON-GAUSSIAN FORM OF THE
            C          PEAKS AND FWHM IS THE LEAST-SQUARE-FITTED VALUE
            C            CORRESPONDING TO THE PEAK ENERGY (TAREA)
C173             WRITE (JPRINT,2081)
C174        2081 FORMAT(1H1, 47X, 14H PEAK ANALYSIS)
C175             WRITE (JPRINT,1003)
C176        1003 FORMAT(1H ,46HNO.  ENERGY   PK CNTR    HEIGHT   H TO BG   ,
                161H AREA(A)    AREA(B)    INT(B)  ERROR(B)    W(A)    W(B)  BASE,
                26H  TYPE)
C177             WRITE (JPRINT,1004)
C178        1004 FORMAT(1H ,46H          KEV    CHAN NO    COUNTS    RATIO    ,
                161H COUNTS      COUNTS            PERCENT      KEV     KEV  CHAN)
            C    IPUNCH LESS THAN ZERO PUNCHES OUT PEAK ANALYSIS DATA
C179             IF (IPUNCH) 1237,1238,1238
C180        1237 CONTINUE
```

```
C181                WRITE (JPUNCH,1236)
C182          1236 FORMAT(44H  NO.  ENERGY(KEV)     INTENSITY     ERROR P/C,
                  126H  AREA(GAUS)     AREA(SUM))
C183          1238 CONTINUE
                C     I IS THE NUMBER OF THE PEAK AS SET BY THE PKLOC SUBROUTINE
                C     N IS THE NUMBER OF THE PEAK PRINTED AND PUNCHED
C184                N = 1
C185                I = 0
                C
                C     LOOP FOR PEAK ANALYSIS
C186           810  CONTINUE
C187                I = I + 1
C188                IF(I-NUMPK) 1856,1856,870
                C     THE ZERO POINTS OF A PEAK ARE CALCULATED AGAIN FOR ALL PEAKS
C189           1856 J = TBL(I+1000) - 1.0
C190            801 IF (IBK(J)) 802,802,803
C191            803 J = J - 1
C192                GO TO 801
C193            802 IA = J
                C     IA IS ZERO POINT LOW ENERGY SIDE.
C194                J = TBL(I+1000) + 1.0
C195            804 IF (IBK(J)) 805,805,806
C196            806 J=J+1
C197                GO TO 804
C198            805 IL=J
                C     IL      IS ZERO POINT , HIGH ENERGY SIDE.
C199                L1 = TBL(I+1000)
C200                L2 = TBL(I+1001)
C201                L3 = TBL(I+1002)
C202                L4 = TBL(I+1003)
C203                XL12=FLOAT (L2-L1)
C204                XL23=FLOAT (L3-L2)
                C     LOGIC TO DETERMINE IF THE PEAK DATA IS IN MULTIPLET FORM.
C205                IF (IL-L2) 811,811,812
C206           812  IF (IL-L3) 813,813,814
C207           814  IF (IL-L4) 815,815,816
                C
                C     SINGLET PEAK CALCULATION
C208             811 CONTINUE
                C     CONDITION TO DETERMINE IF PEAK HAS BEEN USED IN EVALUATING THE
                C          LSF FWHM IN THE PREVIOUS PART OF THE PROGRAM
C209                IF(IBL(I+1600)) 1855,1855,1860
                C
                C     SINGLETS PEAKS WHOSE PARAMETERS HAVE BEEN STORED IN THE TBL ARRAY
C210           1860 EGAM = TBL(I + 1600)
C211                GX = TBL(I + 2200)
C212                WIDTH = TBL(I + 2800)
C213                SAREA = TBL(I + 3400)
```

```
FORTRAN IV G LEVEL 1, MOD 2            PKANAL          DATE = 68223          19/29/33

0214                  PCNR = TBL(I + 420)
0215                  PKHT = 2.0*GX
0216                  EGM = EGAM/1000.
0217                  FWH = SQRT(A0 + A1*EGM + A2*EGM*EGM)
                C     FWH IS THE LSF FWHM OF THE PEAKS AT GIVEN ENERGY
0218                  TAREA = 2.1289*TERA*FWH*GX/SLP
0219                        CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
                C     FIRENG IS THE FIRST ENERGY (KEV) VALUE USED IN EFFICIENCY CALC.
                C     DELE IS THE ENERGY (KEV) DIFFERENCE BETWEEN EFFICIENCY POINTS.
                C     SANGLE IS THE SOLID ANGLE CORRECTION TERM
                C     FLUXT IS THE TOTAL NUMBER OF NEUTRONS INCIDENT ON THE SAMPLE.
                C     AIGAM IS THE PEAK AREA CORRECTED FOR EFFCY. SOLID ANGLE, AND FLUX.
0220                  ER = TBL(I+3750)
                C     ER IS THE PERCENT ERROR INVOLVED IN THE AREA CALCULATION (TAREA)
                C     Z IS THE NUMBER OF CHANNELS OCCUPIED BY THE PEAK
0221                  Z = FLOAT (IL-IA)
                C     HTOB IS THE RATIO OF THE PEAK HEIGHT TO AN APPROXIMATE
                C     VALUE OF THE BACKGROUND UNDER THE PEAK
                C     THE ADDITIONAL 5.0 POINTS ARE TO INSURE THAT HTOB DOES NOT GO
                C     TO INFINITY WHEN THE BACKGROUND GOES TO ZERO (END OF SPECTRUM)
0222                  K = L1/10 + 1
0223                  HTOB = PKHT/(TBL(K) + 5.0)
0224                  WRITE (JPRINT,1006) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                     1WIDTH,FWH,Z
0225            1006 FORMAT(I4,3X,F7.1,3X,F7.1,3X,F7.1,3X,F7.3,3X,F8.1,3X,F8.1,3X,F7.2,
                     13X,F7.2,3X,F5.2,3X,F5.2,2X,F4.0,3X,2H*S)
0226                  IF (IPUNCH) 1239,1240,1240
0227            1239  CONTINUE
0228                  WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0229            1235  FORMAT(I5,5(3X,F10.2))
0230            1240  CONTINUE
0231                  N = N+ 1
0232                  GO TO 810
                C
                C     ANALYSIS OF PEAKS NOT USED IN THE COMPUTATION OF THE LSF FWHM
                C     AREA     IS AREA UNDER THE PEAK.
0233            1855  AREA = 0.0
0234                  DO 8C7 K=IA,IL
0235            8C7   AREA=AREA+TBK(K)
0236                  SAREA = AREA
                C     ORT IS CALCULATED PEAK CENTER.
0237                  ORT  = FLOAT (L1) + DTS(L1)/(DTS(L1-1)-DTS(L1)) + 0.5
0238                  PCNR=ORT
0239                  LY = L1
0240                  PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                     1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0241                  K = L1/10 + 1
0242                  ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
```

```
0243              IF (ER-ECR) 871,871,810
0244            871 CONTINUE
                C     SUBROUTINE ADJUST CORRECTS PCNR TO OBTAIN THE TRUE PEAK CENTER
                C           USING CNTR AND CCRR FROM THE LINEARITY SUBROUTINE.
0245                      CALL ADJUST (CNTR,CCRR,PCNR,TPC)
                C     EGAM    GAMMA ENERGY OF THE PEAK.
0246              EGAM=EGAM1+SLP*(TPC-TPC1)
                C     EVALUATION OF LEAST-SQUARE-FITTED FWHM
0247              EGM = EGAM/1000.0
                C     IF THE CALC. FWHM**2 IS .LE. (FPS-EREW)**2 THEN FWHM = FPS.
                C     THIS CONDITION PREVENTS A POSSIBLE SQRT(NEG.NO.).
0248              FWH = FPS
0249              DUM = AC + A1*EGM + A2*ECM*EGM
0250              DUMM = DUM - (FPS-EREW)*(FPS-EREW)
0251              IF (DUMM.GE.0.0) FWH = SQRT(DUM)
0252              GX = 0.5*PKHT
0253              TAREA = 2.1289*TERA*FWH*GX/SLP
                C     EVALUATION OF ACTUAL FWHM
0254             1854 J = L1
0255              842 IF (TBK(J)-GX) 840,840,841
0256              841 J=J-1
0257                  GO TO 842
                C     PHL     CHANNEL NUMBER LOW ENERGY SIDE.
0258              840 PHL=FLOAT (J)+(GX-TBK(J))/(TBK(J+1)-TBK(J))
0259                  J=L1
0260              848 IF (TBK(J)-GX) 845,845,846
0261              846 J=J+1
0262                  GO TO 848
0263              845 W=FLOAT (J)-(GX-TBK(J))/(TBK(J-1)-TBK(J))-PHL
                C     W       FWHM IN CHANNEL NUMBERS.
                C     WIDTH   FWHM IN ENERGY.
0264                  WIDTH=W*SLP
                C     SUBROUTINE INTSTY CORRECTS THE PEAK AREA FOR EFFICIENCY OF THE
                C     SYSTEM AT ENERGY EGAM.
0265                        CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
                C     Z IS THE NUMBER OF CHANNELS OCCUPIED BY THE PEAK
0266                  Z = FLOAT (IL -IA)
0267                  HTOB = PKHT/(TBL(K) + 5.0)
                C     DATA OUTPUT
0268                  WRITE (JPRINT,1007) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                     1WIDTH,FWH,Z
0269             1007 FORMAT(I4,3X,F7.1,3X,F7.1,3X,F7.1,3X,F7.3,3X,F8.1,3X,F8.1,3X,F7.2,
                     13X,F7.2,3X,F5.2,3X,F5.2,2X,F4.0,3X,2H S)
0270                  IF (IPUNCH) 1241,1242,1242
0271             1241 CONTINUE
0272                  WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0273             1242 CONTINUE
0274                  N = N + 1
```

```
0275                GO TO 810
                C
                C       DOUBLET PEAK CALCULATION
                C       THE PROCEDURE USED HERE IS SIMILAR TO THE SINGLET PEAK CALC.
                C       TWO GAUSSIAN CURVES ARE FITTED TO THE DATA.
0276             813    AREA=0.
0277                    DO 827 K=IA,IL
0278             827    AREA=AREA+TBK(K)
                C       ORT1     IS THE CALCULATED CENTER OF THE FIRST PEAK.
0279                    ORT1 = FLOAT (L1) + DTS(L1)/(DTS(L1-1)-DTS(L1)) + 0.5
                C       ORT2     IS THE CALCULATED CENTER OF THE SECOND PEAK.
0280                    ORT2 = FLOAT (L2) + DTS(L2)/(DTS(L2-1)-DTS(L2)) + 0.5
                C       AN AVERAGE FWHM IS CALCULATED FOR THE MULTIPLET IN ALL CASES
0281                    PCNR = (ORT1 + ORT2)/2.0
0282                    CALL ADJUST (CNTR,CORR,PCNR,TPC)
0283                    EGAM = EGAM1 + SLP*(TPC-TPC1)
0284                    EGM = EGAM/1000.
0285                    FWH = FPS
0286                    DUM = A0 + A1*EGM + A2*EGM*EGM
0287                    DUMM = DUM - (FPS-ERFW)*(FPS-ERFW)
0288                    IF (DUMM.GE.0.0) FWH = SQRT(DUM)
0289                    WIDTH = FWH
0290                    VRNC = (FWH/(SLP*2.35478))**2
0291                    C=EXP (-(XL12**2)/(2.*VRNC))
                C       L1       IS THE CHANNEL NUMBER OF THE MAXIMUM OF THE FIRST PEAK.
                C       L2       IS THE CHANNEL NUMBER OF THE MAXIMUM OF THE SECOND PEAK.
0292                    AREA1=AREA*(TBK(L1)-TBK(L2)*C)/((TBK(L1)+TBK(L2))*(1.-C))
0293                    AREA2=AREA*(TBK(L2)-TBK(L1)*C)/((TBK(L1)+TBK(L2))*(1.-C))
0294                    LY = L1
0295                    DCOE = 1.0645*TERA*FWH/SLP
0296                    DENOM = 1.0 - C**2
0297                    PCNR = ORT1
0298                    PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                       1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0299                    PKHT1 = PKHT
0300                    PCNR = ORT2
0301                    LY = L2
0302                    PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                       1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0303                    PKHT2 = PKHT
                C       FIRST PEAK DATA CALCULATION.
0304                    PKHT = (PKHT1 - C*PKHT2)/DENOM
0305                    IF (PKHT.LE.0.0) GO TO 892
0306                    K = L1/10 + 1
0307                    ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
0308                    IF (ER-ECR) 872,872,892
0309             872    PCNR=ORT1
0310                    TAREA = DCOE*PKHT
```

```
FCRTRAN IV G LEVEL 1, MOD 2          PKANAL          DATE = 68223          19/29/33

0311                  SAREA=AREA1
0312                  IF (SAREA.LE.0.0) GO TO 892
0313                        CALL ADJUST (CNTR,CORR,PCNR,TPC)
0314                  EGAM=EGAM1+SLP*(TPC-TPC1)
0315                        CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
0316                  Z = FLOAT (IL-IA)
0317                  HTOB = PKHT/(TBL(K) + 5.0)
0318                  WRITE (JPRINT,1008) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                     1 WIDTH,FWH,Z,C
0319               1008 FORMAT(I4,3X,F7.1,3X,F7.1,3X,F7.1,3X,F7.3,3X,F8.1,3X,F8.1,3X,F7.2,
                     1 3X,F7.2,3X,F5.2,3X,F5.2,2X,F4.0,' D ',F4.2)
0320                  IF (IPUNCH) 1243,1244,1244
0321               1243 CONTINUE
0322                  WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0323               1244 CONTINUE
0324                  N = N + 1
0325                892  I = I+1
                   C      SECOND PEAK DATA CALCULATION.
0326                  PKHT = (PKHT2 - C*PKHT1)/DENOM
0327                  IF (PKHT.LE.0.0) GO TO 810
0328                  K = L2/10 + 1
0329                  ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
0330                  IF (ER-ECR) 873,873,810
0331               873  PCNR=ORT2
0332                  TAREA = COOE*PKHT
0333                  SAREA=AREA2
0334                  IF (SAREA.LE.0.0) GO TO 810
0335                        CALL ADJUST (CNTR,CORR,PCNR,TPC)
0336                  EGAM=EGAM1+SLP*(TPC-TPC1)
0337                        CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
0338                  Z = FLOAT (IL-IA)
0339                  HTOB = PKHT/(TBL(K) + 5.0)
0340                  WRITE (JPRINT,1008) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                     1 WIDTH,FWH,Z,C
0341                  IF (IPUNCH) 1245,1246,1246
0342               1245 CONTINUE
0343                  WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0344               1246 CONTINUE
0345                  N = N + 1
0346                  GO TO 810
                   C
                   C      TRIPLET PEAK CALCULATION
                   C      THE PROCEDURE USED HERE IS SIMILAR TO THE SINGLET PEAK CALC.
                   C      THREE GAUSSIAN CURVES ARE FITTED TO THE TRIPLET DATA.
                   C      AN AVERAGE FWHM IS CALCULATED FOR THE TRIPLET CASE.
0347                815  AREA=0.
0348                  DO 837 K=IA,IL
0349                837  AREA=AREA+TBK(K)
```

```
0350              ORT1 = FLOAT (L1) + DTS(L1)/(DTS(L1-1)-DTS(L1)) + 0.5
0351              ORT2 = FLOAT (L2) + DTS(L2)/(DTS(L2-1)-DTS(L2)) + 0.5
0352              ORT3 = FLOAT (L3) + DTS(L3)/(DTS(L3-1)-DTS(L3)) + 0.5
0353              PCNR =(ORT1 + ORT2 + ORT3)/3.
0354                    CALL ADJUST (CNTR,CORR,PCNR,TPC)
0355              EGAM = EGAM1 + SLP*(TPC-TPC1)
0356              EGM = EGAM/1000.
0357              FWH = FPS
0358              DUM = A0 + A1*EGM + A2*EGM*EGM
0359              DUMM = DUM - (FPS-EREW)*(FPS-EREW)
0360              IF (DUMM.GE.0.0) FWH = SQRT(DUM)
0361              WIDTH = FWH
0362              VRNC = (FWH/(SLP*2.35478))**2
0363              C=EXP (-(XL12**2)/(2.*VRNC))
0364              D=EXP (-(XL23**2)/(2.*VRNC))
0365              TBSM=TBK(L1)*(1.-D**2-C*(1.-D))+TBK(L2)*(1.-C-D)+TBK(L3)*
                 1(1.-C**2-D*(1.-C))
0366              AREA1=AREA*(TBK(L1)*(1.-D**2)-TBK(L2)*C+TBK(L3)*C*D)/TBSM
0367              AREA2=AREA*(-TBK(L1)*C+TBK(L2)-TBK(L3)*D)/TBSM
0368              AREA3=AREA*(TBK(L1)*C*D-TBK(L2)*D+TBK(L3)*(1.-C**2))/TBSM
0369              TCOE = 1.0645*TERA*FWH/SLP
0370              DENOM = 1.0 - C**2 - D**2
0371              LY = L1
0372              PCNR = ORT1
0373              PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                 1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0374              PKHT1 = PKHT
0375              LY = L2
0376              PCNR = ORT2
0377              PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                 1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0378              PKHT2 = PKHT
0379              LY = L3
0380              PCNR = ORT3
0381              PKHT = TBK(LY) + (PCNR-FLOAT(LY)) * DTS(LY) +
                 1 0.5 * (PCNR-FLOAT(LY))*(PCNR-FLOAT(LY+1))*(DTS(LY)-DTS(LY-1))
0382              PKHT3 = PKHT
                C     FIRST PEAK DATA CALCULATION.
0383              PKHT = (PKHT1*(1.0-D**2)-PKHT2*C+PKHT3*C*D)/DENOM
0384              IF (PKHT.LE.0.0) GO TO 894
0385              K = L1/10 + 1
0386              ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
0387              IF (ER-ECR) 874,874,894
0388            874 PCNR=ORT1
0389              TAREA = TCOE*PKHT
0390              SAREA=AREA1
0391              IF (SAREA.LE.0.0) GO TO 894
0392                    CALL ADJUST (CNTR,CORR,PCNR,TPC)
```

```
0393                EGAM=EGAM1+SLP*(TPC-TPC1)
0394                      CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
0395                Z = FLCAT (IL-IA)
0396                HTOB = PKHT/(TBL(K) + 5.0)
0397                WRITE (JPRINT,1009) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                   1 WIDTH,FWH,Z,C,D
0398             1009 FORMAT(I4,3X,F7.1,3X,F7.1,3X,F7.1,3X,F7.3,3X,F8.1,3X,F8.1,3X,F7.2,
                   1 3X,F7.2,3X,F5.2,3X,F5.2,2X,F4.0,' T ',F4.2,1X,F4.2)
0399                IF (IPUNCH) 1247,1248,1248
0400             1247 CONTINUE
0401                WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0402             1248 CONTINUE
0403                N = N + 1
0404              894 I = I + 1
                 C      SECOND PEAK DATA CALCULATION
0405                PKHT = (-PKHT1*C + PKHT2 - PKHT3*D)/DENOM
0406                IF (PKHT.LE.0.0) GO TO 895
0407                K = L2/10 + 1
0408                ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
0409                IF (ER-ECR) 875,875,895
0410              875  PCNR=ORT2
0411                TAREA = TCOE*PKHT
0412                SAREA=AREA2
0413                IF (SAREA.LE.0.0) GO TO 895
0414                      CALL ADJUST (CNTR,CORR,PCNR,TPC)
0415                EGAM=EGAM1+SLP*(TPC-TPC1)
0416                      CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
0417                Z = FLCAT (IL-IA)
0418                HTOB = PKHT/(TBL(K) + 5.0)
0419                WRITE (JPRINT,1009) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                   1 WIDTH,FWH,Z,C,D
0420                IF (IPUNCH) 1249,1250,1250
0421             1249 CONTINUE
0422                WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
0423             1250 CONTINUE
0424                N=N+1
0425              895 I = I + 1
                 C      THIRD PEAK DATA CALCULATION.
0426                PKHT=(PKHT1*C*D-PKHT2*D+PKHT3*(1.0-C**2))/DENOM
0427                IF (PKHT.LE.0.0) GO TO 810
0428                K = L3/10 + 1
0429                ER = 100.0 * SQRT(2.0*(PKHT+1.5*TBL(K)))/(ERTM*PKHT)
0430                IF (ER-ECR) 876,876,810
0431              876  PCNR=ORT3
0432                TAREA = TCOE*PKHT
0433                SAREA=AREA3
0434                IF (SAREA.LE.0.0) GO TO 810
0435                      CALL ADJUST (CNTR,CORR,PCNR,TPC)
```

```
FORTRAN IV G LEVEL 1, MOD 2          PKANAL          DATE = 68223          19/29/33

C436                EGAM=EGAM1+SLP*(TPC-TPC1)
0437                      CALL INTSTY (TAREA,EGAM,FIRENG,DELENG,SANGLE,FLUXT,AIGAM)
0438                Z = FLOAT (IL-IA)
0439                HTOB = PKHT/(TBL(K) + 5.0)
0440                WRITE (JPRINT,1009) N,EGAM,PCNR,PKHT,HTOB,SAREA,TAREA,AIGAM,ER,
                   1 WIDTH,FWH,Z,C,D
0441                IF (IPUNCH) 1251,1252,1252
0442             1251 CONTINUE
0443                WRITE (JPUNCH,1235) N,EGAM,AIGAM,ER,TAREA,SAREA
C444             1252 CONTINUE
0445                N=N+1
0446                GO TO 810
                 C
                 C      MULTIPLET PEAK ANALYSIS
                 C      MULPIPLETS WHICH ARE SO LABELED BECAUSE OF TINY PEAKS AT THE
                 C      WINGS OF STRONG ONES ARE ANALYSED AS TRIPLETS.  THE THREE
                 C      STRONGEST PEAKS IN THE MULTIPLET ARE CHOSEN FOR THAT PURPOSE.
                 C      THE SMOOTHED DATA SHOULD BE EXAMINED TO INSURE THIS.
0447               816 CONTINUE
0448                   II = 3
0449                   WT(1) = FLOAT(L1)
0450                   WT(2) = FLOAT(L2)
0451                   WT(3)= FLOAT(L3)
C452                   WT(4) = FLOAT(L4)
                 C     EVALUATION OF II, THE NUMBER OF PEAKS IN THE MULTIPLET
0453               817 II = II + 1
0454                   I3 = II + I + 1000
0455                   WT(II+1) = TBL(I3)
0456                   LZ = TBL(I3)
0457                   IF(IL-LZ) 819,819,817
                 C     ORDERING OF THE PEAKS IN TERMS OF DECREASING PEAK HEIGHT
0458               819 DO 3019 IE = 1,II
0459                   LS = WT(IE)
0460                   WT(IE+10) = TBK(LS)
C461              3019 WT(IE+20) = TBK(LS)
                 C     WT(I+20) ARE THE PEAK HEIGHTS IN ORDER THEY APPEAR.
                 C     WT(I+10 ARE THE VALUES OF THE PEAK HEIGHTS ORDERED BY
                 C     DECREASING VALUE.
0462                   DO 3020 IP = 1,II
0463                   DO 3020 IE = IP,II
C464                   IF (WT(IP+10) - WT(IE+10)) 3021,3020,3020
0465              3021 ROK = WT(IE+10)
0466                   WT(IE+10) = WT(IP+10)
0467                   WT(IP+10) = ROK
0468              3020 CONTINUE
0469                   NR = 0
C470                   DO 3025 NP=1,3
0471                   DO 3026 NQ=1,II
```

```
FORTRAN IV G LEVEL 1, MOD 2            PKANAL            DATE = 68223          19/29/33

0472                IF (WT(NP+10) - WT(NQ+20)) 3026,3027,3026
0473          3027  NR = NR + 1
0474                IV = I + 999 + NQ
0475                WT(NR) = TBL(IV)
0476                GO TO 3025
0477          3026  CONTINUE
0478          3025  CONTINUE
0479                L1 = WT(1)
0480                L2 = WT(2)
0481                L3 = WT(3)
0482                XL12 = FLOAT (L2-L1)
0483                XL23 = FLOAT (L3-L2)
0484                WRITE (JPRINT,820) II
0485          820   FORMAT (1H ,23X,27H NUMBER OF MULTIPLE PEAKS =,I5)
0486                WRITE (JPRINT,3033)
0487          3033  FORMAT(1H ,49H THE 3 STRONGEST PEAKS ARE ANALYSED AS A TRIPLET.,
                   134H CHECK BACKGROUND SUBTRACTED DATA.)
0488                I = I + II - 3
0489                GO TO 815
                C
0490          870   CONTINUE
0491                RETURN
0492                END
```

```
C                                                        CTFFT
C     THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C     TRANSFORM(J1,J2,...) = SUM(DATA(I1,I2,...)*W1**((I1-1)*(J1-1))
C                               *W2**((I2-1)*(J2-1))*...),
C     WHERE I1 AND J1 RUN FROM 1 TO NN(1) AND W1=EXP(ISIGN*2*PI*
C     SQRT(-1)/NN(1)), ETC.  THERE IS NO LIMIT ON THE DIMENSIONALITY
C     (NUMBER OF SUBSCRIPTS) OF THE DATA ARRAY.  IF AN INVERSE
C     TRANSFORM (ISIGN=+1) IS PERFORMED UPON AN ARRAY OF TRANSFORMED
C     (ISIGN=-1) DATA, THE ORIGINAL DATA WILL REAPPEAR,
C     MULTIPLIED BY NN(1)*NN(2)*...  THE ARRAY OF INPUT DATA MAY BE
C     REAL OR COMPLEX, AT THE PROGRAMMERS OPTION, WITH A SAVING OF
C     UP TO FORTY PER CENT IN RUNNING TIME FOR REAL OVER COMPLEX.
C     (FOR FASTEST TRANSFORM OF REAL DATA, NN(1) SHOULD BE EVEN.)
C     THE TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED IN THE
C     ORIGINAL ARRAY OF DATA, REPLACING THE INPUT DATA.  THE LENGTH
C     OF EACH DIMENSION OF THE DATA ARRAY MAY BE ANY INTEGER.  THE
C     PROGRAM RUNS FASTER ON COMPOSITE INTEGERS THAN ON PRIMES, AND IS
C     PARTICULARLY FAST ON NUMBERS RICH IN FACTORS OF TWO.
C     TIMING IS IN FACT GIVEN BY THE FOLLOWING FORMULA.  LET NTOT BE THE
C     TOTAL NUMBER OF POINTS (REAL OR COMPLEX) IN THE DATA ARRAY, THAT
C     IS, NTOT=NN(1)*NN(2)*...  DECOMPOSE NTOT INTO ITS PRIME FACTORS,
C     SUCH AS 2**K2 * 3**K3 * 5**K5 * ...  LET SUM2 BE THE SUM OF ALL
C     THE FACTORS OF TWO IN NTOT, THAT IS, SUM2 = 2*K2.  LET SUMF BE
C     THE SUM OF ALL OTHER FACTORS OF NTOT, THAT IS, SUMF = 3*K3+5*K5+...
C     THE TIME TAKEN BY A MULTIDIMENSIONAL TRANSFORM ON THESE NTOT DATA
C     IS T = TO + NTOT*(T1+T2*SUM2+T3*SUMF).  ON THE CDC 3300 (FLOATING
C     POINT ADD TIME = SIX MICROSECONDS), T = 3000 + NTOT*(600+40*SUM2+
C     175*SUMF) MICROSECONDS ON COMPLEX DATA.
C
C     IMPLEMENTATION OF THE DEFINITION BY SUMMATION WILL RUN IN A TIME
C     PROPORTIONAL TO NTOT*(NN(1)+NN(2)+...).  FOR HIGHLY COMPOSITE NTOT
C     THE SAVINGS OFFERED BY THIS PROGRAM CAN BE DRAMATIC.  A ONE-DIMEN-
C     SIONAL ARRAY 4000 IN LENGTH WILL BE TRANSFORMED IN 4000*(600+
C     40*(2+2+2+2+2)+175*(5+5+5)) = 14.5 SECONDS VERSUS ABOUT 4000*
C     4000*175 = 2800 SECONDS FOR THE STRAIGHTFORWARD TECHNIQUE.
C
C     THE CALLING SEQUENCE IS--
C     CALL FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)
C
C     DATA IS THE ARRAY USED TO HOLD THE REAL AND IMAGINARY PARTS
C     OF THE DATA ON INPUT AND THE TRANSFORM VALUES ON OUTPUT.  IT
C     IS A MULTIDIMENSIONAL FLOATING POINT ARRAY, WITH THE REAL AND
C     IMAGINARY PARTS OF A DATUM STORED IMMEDIATELY ADJACENT IN STORAGE
C     (SUCH AS FORTRAN IV PLACES THEM).  THE EXTENT OF EACH DIMENSION
C     IS GIVEN IN THE INTEGER ARRAY NN, OF LENGTH NDIM.  ISIGN IS -1
C     TO INDICATE A FORWARD TRANSFORM (EXPONENTIAL SIGN IS -) AND +1
C     FOR AN INVERSE TRANSFORM (SIGN IS +).  IFORM IS +1 IF THE DATA AND
C     THE TRANSFORM VALUES ARE COMPLEX.  IT IS 0 IF THE DATA ARE REAL
C     BUT THE TRANSFORM VALUES ARE COMPLEX.  IF IT IS 0, THE IMAGINARY
C     PARTS OF THE DATA SHOULD BE SET TO ZERO.  AS EXPLAINED ABOVE, THE
C     TRANSFORM VALUES ARE ALWAYS COMPLEX AND ARE STORED IN ARRAY DATA.
C     WORK IS AN ARRAY USED FOR WORKING STORAGE.  IT IS NOT NECESSARY
C     IF ALL THE DIMENSIONS OF THE DATA ARE POWERS OF TWO.  IN THIS CASE
C     IT MAY BE REPLACED BY 0 IN THE CALLING SEQUENCE.  THUS, USE OF
C     POWERS OF TWO CAN FREE A GOOD DEAL OF STORAGE.  IF ANY DIMENSION
C     IS NOT A POWER OF TWO, THIS ARRAY MUST BE SUPPLIED.  IT IS
C     FLOATING POINT, ONE DIMENSIONAL OF LENGTH EQUAL TO TWICE THE
C     LARGEST ARRAY DIMENSION (I.E., NN(I) ) THAT IS NOT A POWER OF
```

```
C     TWO.  THEREFORE, IN ONE DIMENSION FOR A NON POWER OF TWO,
C     WORK OCCUPIES AS MANY STORAGE LOCATIONS AS DATA.  IF SUPPLIED,
C     WORK MUST NOT BE THE SAME ARRAY AS DATA.  ALL SUBSCRIPTS OF ALL
C     ARRAYS BEGIN AT ONE.
C
C     THE FAST FOURIER ALGORITHM PLACES TWO RESTRICTIONS UPON THE
C     NATURE OF THE DATA BEYOND THE USUAL RESTRICTION THAT
C     THE DATA FORM ONE CYCLE OF A PERIODIC FUNCTION.  THEY ARE--
C     1) THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES
C     MUST BE THE SAME.
C     2) CONSIDERING THE DATA TO BE IN THE TIME DOMAIN,
C     THEY MUST BE EQUI-SPACED AT INTERVALS OF DT.  FURTHER, THE TRANS-
C     FORM VALUES, CONSIDERED TO BE IN FREQUENCY SPACE, WILL BE EQUI-
C     SPACED FROM 0 TO 2*PI*(NN(I)-1)/(NN(I)*DT) AT INTERVALS OF
C     2*PI/(NN(I)*DT) FOR EACH DIMENSION OF LENGTH NN(I).  OF COURSE,
C     DT NEED NOT BE THE SAME FOR EVERY DIMENSION.
C
C     THERE ARE NO ERROR MESSAGES OR ERROR HALTS IN THIS PROGRAM.  THE
C     PROGRAM RETURNS IMMEDIATELY IF NDIM OR ANY NN(I) IS LESS THAN ONE.
C     EXAMPLE 1.  THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A
C     COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV.
C     DIMENSION DATA(32,25,13),WORK(50),NN(3)
C     COMPLEX DATA
C     DATA NN/32,25,13/
C     DO 1 I=1,32
C     DO 1 J=1,25
C     DO 1 K=1,13
C   1 DATA(I,J,K)=COMPLEX VALUE
C     CALL FOURT(DATA,NN,3,-1,1,WORK)
C
C     EXAMPLE 2.  ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF
C     LENGTH 64 IN FORTRAN II.
C     DIMENSION DATA(2,64)
C     DO 2 I=1,64
C     DATA(1,I)=REAL PART
C   2 DATA(2,I)=0.
C     CALL FOURT(DATA,64,1,-1,0,0)
C
C     FOR THE DIGIT REVERSAL WAS SUGGESTED BY RALPH ALTER (ALSO MIT LL).
C     THIS IS THE FASTEST AND MOST VERSATILE VERSION OF THE FFT KNOWN
C     TO THE AUTHOR.  A PROGRAM CALLED FOUR2 IS AVAILABLE THAT ALSO
C     PERFORMS THE FAST FOURIER TRANSFORM AND IS WRITTEN IN USASI BASIC
C     FORTRAN.  IT IS ABOUT ONE THIRD AS LONG AND RESTRICTS THE
C     DIMENSIONS OF THE INPUT ARRAY (WHICH MUST BE COMPLEX) TO BE POWERS
C     OF TWO.  ANOTHER PROGRAM, CALLED FOUR1, IS ONE TENTH AS LONG AND
C     RUNS TWO THIRDS AS FAST ON A ONE-DIMENSIONAL COMPLEX ARRAY WHOSE
C     LENGTH IS A POWER OF TWO.
C
C     REFERENCE--
C     FAST FOURIER TRANSFORMS FOR FUN AND PROFIT, W. GENTLEMAN AND
C     G. SANDE, 1966 FALL JOINT COMPUTER CONFERENCE.
C
C     THE WORK REPORTED IN THIS DOCUMENT WAS PERFORMED AT LINCOLN LAB-
C     ORATORY, A CENTER FOR RESEARCH OPERATED BY MASSACHUSETTS INSTITUTE
C     OF TECHNOLOGY, WITH THE SUPPORT OF THE U.S. AIR FORCE UNDER
C     CONTRACT AF 19(628)-5167.
C
```

```
C       PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES
C       RADER (BOTH OF MIT LINCOLN LABORATORY), MAY 1967.  THE IDEA
        SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)
C       THE FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C
        DIMENSION DATA(1),NN(1),IFACT(32),WORK(1)
        TWOPI=6.283185307
        RTHLF=.70710 67812
        IF(NDIM-1)920,1,1
1       NTOT=2
        DO 2 IDIM=1,NDIM
        IF(NN(IDIM))920,920,2
2       NTOT=NTOT*NN(IDIM)
0
C       MAIN LOOP FOR EACH DIMENSION
C
        NP1=2
        DO 910 IDIM=1,NDIM
        N=NN(IDIM)
        NP2=NP1*N
        IF(N-1)920,900,5
C
C       IS N A POWER OF TWO AND IF NOT, WHAT ARE ITS FACTORS
C
5       M=N
        NTWO=NP1
        IF=1
        IDIV=2
10      IQUOT=M/IDIV
        IREM=M-IDIV*IQUOT
        IF(IQUOT-IDIV)50,11,11
11      IF(IREM)20,12,20
12      NTWO=NTWO+NTWO
        IFACT(IF)=IDIV
        IF=IF+1
        M=IQUOT
        GO TO 10
20      IDIV=3
        INON2=IF
30      IQUOT=M/IDIV
        IREM=M-IDIV*IQUOT
        IF(IQUOT-IDIV)60,31,31
31      IF(IREM)40,32,40
32      IFACT(IF)=IDIV
        IF=IF+1
        M=IQUOT
        GO TO 30
40      IDIV=IDIV+2
        GO TO 30
50      INON2=IF
        IF(IREM)60,51,60
51      NTWO=NTWO+NTWO
        GO TO 70
60      IFACT(IF)=M
70      NON2P=NP2/NTWO
C
C       SEPARATE FOUR CASES--
```

```
C         1. COMPLEX TRANSFORM
C         2. REAL TRANSFORM FOR THE 2ND, 3RD, ETC. DIMENSION.  METHOD--
C            TRANSFORM HALF THE DATA, SUPPLYING THE OTHER HALF BY CON-
C            JUGATE SYMMETRY.
C         3. REAL TRANSFORM FOR THE 1ST DIMENSION, N ODD.  METHOD--
C            SET THE IMAGINARY PARTS TO ZERO.
C         4. REAL TRANSFORM FOR THE 1ST DIMENSION, N EVEN.  METHOD--
C            TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 WHOSE REAL PARTS
C            ARE THE EVEN NUMBERED REAL VALUES AND WHOSE IMAGINARY PARTS
C            ARE THE ODD NUMBERED REAL VALUES.  SEPARATE AND SUPPLY
C            THE SECOND HALF BY CONJUGATE SYMMETRY.
C
      ICASE=1
      IFMIN=1
      I1RNG=NP1
      IF(IDIM-4)74,100,100
74    IF(IFORM)71,71,100
71    ICASE=2
      I1RNG=NP0*(1+NPREV/2)
      IF(IDIM-1)72,72,100
72    ICASE=3
      I1RNG=NP1
      IF(NTWO-NP1)100,100,73
73    ICASE=4
      IFMIN=2
      NTWO=NTWO/2
      N=N/2
      NP2=NP2/2
      NTOT=NTOT/2
      I=1
      DO 80 J=1,NTOT
      DATA(J)=DATA(I)
80    I=I+2
C
C     SHUFFLE DATA BY BIT REVERSAL, SINCE N=2**K.  AS THE SHUFFLING
C     CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED
C
100   IF(NON2P-1)101,101,200
101   NP2HF=NP2/2
      J=1
      DO 150 I2=1,NP2,NP1
      IF(J-I2)121,130,130
21    I1MAX=I2+NP1-2
      DO 125 I1=I2,I1MAX,2
      DO 125 I3=I1,NTOT,NP2
      J3=J+I3-I2
      TEMPR=DATA(I3)
      TEMPI=DATA(I3+1)
      DATA(I3)=DATA(J3)
      DATA(I3+1)=DATA(J3+1)
      DATA(J3)=TEMPR
125   DATA(J3+1)=TEMPI
130   M=NP2HF
140   IF(J-M)150,150,141
141   J=J-M
      M=M/2
      IF(M-NP1)150,140,140
```

```
150   J=J+M
      GO TO 300
C
C     SHUFFLE DATA BY DIGIT REVERSAL FOR GENERAL N
C
200   NWORK=2*N
      DO 270 I1=1,NP1,2
      DO 270 I3=I1,NTOT,NP2
      J=I3
      DO 260 I=1,NWORK,2
      IF(ICASE-3)210,220,210
210   WORK(I)=DATA(J)
      WORK(I+1)=DATA(J+1)
      GO TO 240
220   WORK(I)=DATA(J)
      WORK(I+1)=0.
240   IFP2=NP2
      IF=IFMIN
250   IFP1=IFP2/IFACT(IF)
      J=J+IFP1
      IF(J-I3-IFP2)260,255,255
255   J=J-IFP2
      IFP2=IFP1
      IF=IF+1
      IF(IFP2-NP1)260,260,250
260   CONTINUE
      I2MAX=I3+NP2-NP1
      I=1
      DO 270 I2=I3,I2MAX,NP1
      DATA(I2)=WORK(I)
      DATA(I2+1)=WORK(I+1)
270   I=I+2
C
C     MAIN LOOP FOR FACTORS OF TWO.
C     W=EXP(ISIGN*2*PI*SQRT(-1)*M/(4*MMAX)).  CHECK FOR W=ISIGN*SQRT(-1)
C     AND REPEAT FOR W=W*(1+ISIGN*SQRT(-1))/SQRT(2).
C
300   IF(NTWO-NP1)600,600,305
305   NP1TW=NP1+NP1
      IPAR=NTWO/NP1
310   IF(IPAR-2)350,330,320
320   IPAR=IPAR/4
      GO TO 310
330   DO 340 I1=1,I1RNG,2
      DO 340 K1=I1,NTOT,NP1TW
      K2=K1+NP1
      TEMPR=DATA(K2)
      TEMPI=DATA(K2+1)
      DATA(K2)=DATA(K1)-TEMPR
      DATA(K2+1)=DATA(K1+1)-TEMPI
      DATA(K1)=DATA(K1)+TEMPR
340   DATA(K1+1)=DATA(K1+1)+TEMPI
350   MMAX=NP1
360   IF(MMAX-NTWO/2)370,600,600
370   LMAX=MAXO(NP1TW,MMAX/2)
      DO 570 L=NP1,LMAX,NP1TW
      M=L
```

```
      IF(MMAX-NP1)420,420,380
380   THETA=-TWOPI*FLOAT(LI/FLOAT(4*MMAX)
      IF(ISIGN)400,390,390
390   THETA=-THETA
400   WR=COS(THETA)
      WI=SIN(THETA)
410   W2R=WR*WR-WI*WI
      W2I=2.*WR*WI
     W3R=W2R*WR-W2I*WI
      W3I=W2R*WI+W2I*WR
420   DO 530 I1=1,I1RNG,2
      KMIN=I1+IPAR*M
      IF(MMAX-NP1)430,430,440
430   KMIN=I1
440   KDIF=IPAR*MMAX
450   KSTEP=4*KDIF
   IF(KSTEP-NTWO)460,460,530
460   DO 520 K1=KMIN,NTOT,KSTEP
      K2=K1+KDIF
      K3=K2+KDIF
    K4=K3+KDIF
      IF(MMAX-NP1)470,470,480
470   U1R=DATA(K1)+DATA(K2)
      U1I=DATA(K1+1)+DATA(K2+1)
      U2R=DATA(K3)+DATA(K4)
      U2I=DATA(K3+1)+DATA(K4+1)
      U3R=DATA(K1)-DATA(K2)
      U3I=DATA(K1+1)-DATA(K2+1)
    IF(ISIGN)471,472,472
471   U4R=DATA(K3+1)-DATA(K4+1)
      U4I=DATA(K4)-DATA(K3)
      GO TO 510
472   U4R=DATA(K4+1)-DATA(K3+1)
      U4I=DATA(K3)-DATA(K4)
      GO TO 510
480   T2R=W2R*DATA(K2)-W2I*DATA(K2+1)
      T2I=W2R*DATA(K2+1)+W2I*DATA(K2)
      T3R=WR*DATA(K3)-WI*DATA(K3+1)
      T3I=WR*DATA(K3+1)+WI*DATA(K3)
      T4R=W3R*DATA(K4)-W3I*DATA(K4+1)
      T4I=W3R*DATA(K4+1)+W3I*DATA(K4)
      U1R=DATA(K1)+T2R
      U1I=DATA(K1+1)+T2I
      U2R=T3R+T4R
      U2I=T3I+T4I
      U3R=DATA(K1)-T2R
      U3I=DATA(K1+1)-T2I
      IF(ISIGN)490,500,500
490   U4R=T3I-T4I
      U4I=T4R-T3R
      GO TO 510
500   U4R=T4I-T3I
      U4I=T3R-T4R
510   DATA(K1)=U1R+U2R
      DATA(K1+1)=U1I+U2I
      DATA(K2)=U3R+U4R
      DATA(K2+1)=U3I+U4I
```

```
      DATA(K3)=U1R-U2R
      DATA(K3+1)=U1I-U2I
      DATA(K4)=U3R-U4R
520   DATA(K4+1)=U3I-U4I
      KDIF=KSTEP
      KMIN=4*(KMIN-I1)+I1
      GO TO 450
530   CONTINUE
      M=M+LMAX
      IF(M-MMAX)540,540,570
540   IF(ISIGN)550,560,560
0   TEMPR=WR
      WR=(WR+WI)*RTHLF
      WI=(WI-TEMPR)*RTHLF
      GO TO 410
560   TEMPR=WR
      WR=(WR-WI)*RTHLF
      WI=(TEMPR+WI)*RTHLF
      GO TO 410
570   CONTINUE
      IPAR=3-IPAR
      MMAX=MMAX+MMAX
      GO TO 360
C
C     MAIN LOOP FOR FACTORS NOT EQUAL TO TWO.
C     W=EXP(ISIGN*2*PI*SQRT(-1)*(J1*J2-I3-1)/IFP2)
C
0   IF(NON2P-1)700,700,601
601   IFP1=NTWO
      IF=INON2
610   IFP2=IFACT(IF)*IFP1
   THETA=-TWOPI/FLOAT(IFACT(IF))
      IF(ISIGN)612,611,611
611   THETA=-THETA
612   WSTPR=COS(THETA)
    WSTPI=SIN(THETA)
      DO 650 J1=1,IFP1,NP1
      THETM=-TWOPI*FLOAT(J1-1)/FLOAT(IFP2)
      IF(ISIGN)614,613,613
613   THETM=-THETM
614   WMINR=COS(THETM)
      WMINI=SIN(THETM)
      I1MAX=J1+IIRNG-2
      DO 650 I1=J1,I1MAX,2
      DO 650 I3=I1,NTOT,NP2
      I=1
      WR=WMINR
      WI=WMINI
      J2MAX=I3+IFP2-IFP1
      DO 640 J2=I3,J2MAX,IFP1
      TWOWR=WR+WR
      J3MAX=J2+NP2-IFP2
      DO 630 J3=J2,J3MAX,IFP2
      JMIN=J3-J2+I3
      J=JMIN+IFP2-IFP1
      SR=DATA(J)
      SI=DATA(J+1)
```

```
      OLDSR=0.
      OLDSI=0.
      J=J+IFP1
620   STMPR=SR
      STMPI=SI
      SR=TWOWR*SR-OLDSR+DATA(J)
      SI=TWOWR*SI-OLDSI+DATA(J+1)
      OLDSR=STMPR
     OLDSI=STMPI
      J=J-IFP1
      IF(J-JMIN)621,621,620
1   WORK(I)=WR*SR-WI*SI-OLDSR+DATA(J)
     WORK(I+1)=WI*SR+WR*SI-OLDSI+DATA(J+1)
630   I=I+2
      WTEMP=WR*WSTPI
     WR=WR*WSTPR-WI*WSTPI
640   WI=WI*WSTPR+WTEMP
      I=1
      DO 650 J2=I3,J2MAX,IFP1
      J3MAX=J2+NP2-IFP2
      DO 650 J3=J2,J3MAX,IFP2
      DATA(J3)=WORK(I)
      DATA(J3+I)=WORK(I+1)
50    I=I+2
     IF=IF+1
      IFP1=IFP2
      IF(IFP1-NP2)610,700,700

     COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON-
C     JUGATE SYMMETRIES.
C
700   GO TO (900,800,900,701),ICASE
701   NHALF=N
      N=N+N
      THETA=-TWOPI/FLOAT(N)
      IF(ISIGN)703,702,702
702   THETA=-THETA
703   WSTPR=COS(THETA)
      WSTPI=SIN(THETA)
      WR=WSTPR
    WI=WSTPI
      IMIN=3
      JMIN=2*NHALF-1
      GO TO 725
710   J=JMIN
      DO 720 I=IMIN,NTOT,NP2
      SUMR=(DATA(I)+DATA(J))/2.
      SUMI=(DATA(I+1)+DATA(J+1))/2.
     DIFR=(DATA(I)-DATA(J))/2.
      DIFI=(DATA(I+1)-DATA(J+1))/2.
      DIFI=(DATA(I+1)-DATA(J+1))/2.
      TEMPR=WR*SUMI+WI*DIFR
      TEMPI=WI*SUMI-WR*DIFR
      DATA(I)=SUMR+TEMPR
      DATA(I+1)=DIFI+TEMPI
      DATA(J)=SUMR-TEMPR
      DATA(J+1)=-DIFI+TEMPI
```

```
720     J=J+NP2
        IMIN=IMIN+2
        JMIN=JMIN-2
        WTEMP=WR*WSTPI
        WR=WR*WSTPR-WI*WSTPI
        WI=WI*WSTPR+WTEMP
725     IF(IMIN-JMIN)710,730,740
730     IF(ISIGN)731,740,740
731     DO 735 I=IMIN,NTOT,NP2
735     DATA(I+1)=-DATA(I+1)
740     NP2=NP2+NP2
        NTOT=NTOT+NTOT
        J=NTOT+1
        IMAX=NTOT/2+1
745     IMIN=IMAX-2*NHALF
        I=IMIN
        GO TO 755
750     DATA(J)=DATA(I)
        DATA(J+1)=-DATA(I+1)
755     I=I+2
        J=J-2
        IF(I-IMAX)750,760,760
760     DATA(J)=DATA(IMIN)-DATA(IMIN+1)
        DATA(J+1)=0.
        IF(I-J)770,780,780
765     DATA(J)=DATA(I)
        DATA(J+1)=DATA(I+1)
770     I=I-2
        J=J-2
        IF(I-IMIN)775,775,765
775     DATA(J)=DATA(IMIN)+DATA(IMIN+1)
        DATA(J+1)=0.
        IMAX=IMIN
        GO TO 745
780     DATA(1)=DATA(1)+DATA(2)
        DATA(2)=0.
        GO TO 900
C
C       COMPLETE A REAL TRANSFORM FOR THE 2ND, 3RD, ETC. DIMENSION BY
C       CONJUGATE SYMMETRIES.
C
800     IF(I1RNG-NP1)805,900,900
805     DO 860 I3=1,NTOT,NP2
        I2MAX=I3+NP2-NP1
        DO 860 I2=I3,I2MAX,NP1
        IMAX=I2+NP1-2
        IMIN=I2+I1RNG
        JMAX=2*I3+NP1-IMIN
        IF(I2-I3)820,820,810
810     JMAX=JMAX+NP2
820     IF(IDIM-2)850,850,830
830     J=JMAX+NP0
        DO 840 I=IMIN,IMAX,2
        DATA(I)=DATA(J)
        DATA(I+1)=-DATA(J+1)
840     J=J-2
850     J=JMAX
```

```
      DO 860 I=IMIN,IMAX,NP0
      DATA(I)=DATA(J)
      DATA(I+1)=DATA(J+1)
860   J=J-NP0
C
C     END OF LOOP ON EACH DIMENSION
C
900   NP0=NP1
      NP1=NP2
910   NPREV=N
920   RETURN
      END
```

APPENDIX B

List of GAMANL input. Table B.1 lists the cards in the order in which they appear in the data deck. The parameters are indicated and the formats used are shown below the parameters. Cards labeled (2)b and (7) represent a group of data cards giving the Linearity and Gamma Spectra Data, respectively. For a 4096 channel spectrum, 512 data cards are required to read in the Gamma Spectra Data.

Table B.2 explains each of the input parameters in greater detail, and Table B.3 shows a typical input deck.

TABLE B.1

(1)	Variable	LR, NOCHAN, N2, N3, XN, LIN
	FORMAT	(4I5, F5.0, I5)
(2a)	for LIN.LE. zero	
	Variable	Previously Calc. Linearity Correction data, CNTR and CORR arrays
	FORMAT	(19X, F7.2, 9X, F7.2)
(2b)	for LIN.GT. zero	
	Variable	Linearity Data, TBL array
	FORMAT	((7X, 7(F6.0, 1X))/(8 (F6.0, 1X)))
(3)	Variable	S1G1, MWTL01, WTC1, WTF1, S1G2, MWTL02, WTC2, WTF2
	FORMAT	(F5.0, I5, 2F5.2, F5.0, I5, 2F5.2)
(4)	Variable	J2, FIRENG, DELENG
	FORMAT	(I5, 2F5.0)
(5)	Variable	Efficiency Data Array, EFFCY.
	FORMAT	(7 (E10.3))
(6)	Variable	NUMRUN, NOCHAN, IMAX, DCR, ECR, BGER, IPUNCH, ERTM, SANGLE, FLUXT
	FORMAT	(3I5, F5.0, 2F5.1, I5, F5.0, 2E10.4)
(7)	Variable	Gamma Spectra Data, TBK array
	FORMAT	((7X, 7(F6.0, 1X))/(8 (F6.0, 1X)))
(8)	Variable	EGAM1, IP1, EGAM2, IP2, FPS, ERFW
	FORMAT	(4X, F6.1, 4X, I6, 4X, F6.1, 4X, I6, 3X, F7.2, 3X, F7.2)
(9)	Variable	MO
	FORMAT	(I5)

TABLE B.2

Card	Variable	Function
(1)	LR	Linearity run number
	NOCHAN	Number of channels in linearity run, generally 4095
	N2	First calibration peak for linearity data, if there are 100 peaks N2 = 20
	N3	Second calibration peak for linearity data, if there are 100 peaks N3 = 80
	XN	Background limit for linearity data, peaks must have counts greater than XN in order to be analysed
	LIN	LIN.LEQ.0 uses previously calculated linearity data, in which case NOCHAN is the number of peaks being read in under FORMAT (19X, F7.2, 9X, F7.2) LIN.GTR.0 calculates linearity correction factors using subroutine LINEAR
(2a)	CNTR CORR	CNTR(I) is channel number at which correction CORR(I) is to be applied to correct for system nonlinearity
(2b)	TBL	Linearity Data Deck TBL(I),I=1, NOCHAN
(3) →	S1G1	The width of the Gaussian used in the smoothing calculation. Typical value S1G1 = 128
→	MWTL01	Cutoff channel at which the smoothing weighting function begins to decrease from WTC1 + WTF1 MWTL01 = 512
→	WTC1	The constant term in the smoothing function WTC1 = 0.0
→	WTF1	The varying term in the smoothing function WTF1 = 1.00
→	S1G2	The width of the Gaussian used in the improved resolution calculation Typical value S1G1 = 128.
→	MWTL02	Channel about which the imp. res. weighting function deviates from WTC2. MWTL02 = 412
→	WTC2	Constant term in imp. res. function WTC2 = 0.90
→	WTF2	Varying term indicative of how much imp. res. is desired WTF2 = 2.00
(4)	J2	Number of points in EFFCY, efficiency array. J2 = 19
	FIRENG	First energy value of EFFCY data (in keV) FIRENG = 1000

Card	Variable	Function
	DELENG	Difference in energy between EFFCY points (in keV) DELENG = 500
(5)	EFFCY	Data represents efficiency data at equally spaced energy points. Obtained from experimental results
(6)	NUMRUN	Run Number of Data Spectra
	NOCHAN	Number of Data Points 2^N-1
	IMAX	Channel Number at which analysis is to begin
	DCR	Slope criterion used in BAKSUB to determine if multiplet peaks are present
	ECR	Error criterion used in PKANAL, peaks with percent errors in area greater than ECR are not analysed
	BGER	Background factor for peak height. Peaks with heights less than or equal to BGER X SQRT(background) + 10.0 are not analysed.
	IPUNCH	If < 0 Peak analysis data from PKANAL is punched out. If = 0 no punched output If = 1 smoothed data punched out If = 2 smoothed, background subtracted data punched out If = 3 improved resolution data punched out.
	ERTM	Correction factor, used in error term calculation, due to smoothing
	SANGLE	Solid angle factor used in INTSTY
	FLUXT	Flux, cross-section, time, and misc. correction factor used in INTSTY
(7)	TBK	Data Deck TBK(I), I = 1, NOCHAN
(8)	EGAM1	Energy (in keV) of first calibration peak. EGAM1 0 deletes the peak analysis
	IP1	Peak center of first calibration peak. Program starts search for calibration peak at IP1-4 channels. Also need TBK(IP1) to be greater than the half-maximum of the calibration peak height
	EGAM2	Energy (keV) of second calibration peak.
	IP2	Peak center of second calibration peak.

Card	Variable	Function
	FPS	First peak FWHM (keV), used in PKANAL to fit the FWHM. FPS.LE. zero implies FPS = FWHM of first calibration peak.
	ERFW	Allowable range for FWHM to vary from FPS.
(9)	MO	MO.LE.0 or MO.GT.4 ends the program
		MO = 1 loops to 2001 to accept more data
		MO = 2 loops to 2002 to accept more data
		MO = 3 loops to 2003 to accept more data
		MO = 4 loops to 2004 to accept more data

TABLE B.3

```
C      SAMPLE GAMANL INPUT DECK

//G.SYSIN  DD *
 393   93   20   80   10    0
          1          132.26          -8.94
          2          173.35          -6.94
          3          214.07          -4.56
          4          255.44          -2.84

C      LINEARITY CORRECTION FACTORS FOR 93 PEAKS, AS PREVIOUSLY CALC.

         90          3956.23          2.47
         91          3999.06          2.75
         92          4041.72          3.17
         93          4084.49          3.50
 128. 1024 0.00 1.00 128.  896 1.00 0.40
   191000. 500.
0.100E-05 0.660E-04 0.220E-03 0.400E-03 0.580E-03 0.760E-03 0.890E-03
 0.970E-03 0.970E-03 0.960E-03 0.920E-03 0.850E-03 0.770E-03 0.690E-03
 0.610E-03 0.530E-03 0.440E-03 0.330E-03 0.220E-03
  701 4095  150  0.5 20.0  2.0    0 1.69 0.576E-05 6.000E.11
000001 000010 000010 000010 000010 000010 000010 000010 000007
000024 000018 000018 000010 000017 000020 000018 000017 000015
000019 000019 000007 000021 000011 000019 000020 000036 000023
000030 000020 000037 000033 000033 000042 000060 000057 000031

C      GAMMA SPECTRA DATA DECK CHANNELS 1 TO 4095

000017 000014 000019 000017 000015 000020 000022 000016 004071
000016 000021 000018 000027 000015 000017 000019 000006 004079
000022 000024 000020 000021 000014 000014 000022 000010 004087
000017 000011 000016 000015 000019 000021 000017 000015 004095
  2223.3        563     7367.7        2963        7.20        3.00
   0
/*
```

APPENDIX C

Output from GAMANL, PKANAL subroutine, listing gamma spectral peaks and results. The data recorded was from an iron sample irradiation in the M.I.T. Triple-Coincidence γ-Spectrometer (1).

PEAKS USED IN COMPUTING LSF OF FWHM

NUMBER	PEAK NO.	ENERGY KEV	WIDTH KEV	WEIGHT	PK HEIGHT	AREA/GAUSSIAN
1	2	1613.61	7.211	9.248	578.19	1.1088
2	4	1725.10	7.222	21.574	926.10	1.2583
3	9	2129.56	5.428	2.265	192.62	0.9447
4	17	2470.19	5.925	2.378	212.77	0.9686
5	20	2682.34	7.200	2.021	250.63	0.9378
6	23	2721.50	7.419	12.280	736.24	1.0275
7	25	2835.21	7.161	4.763	406.33	1.0705
8	26	2873.74	7.018	1.725	225.05	1.1164
9	29	2955.49	8.303	1.603	254.20	1.3125
10	32	3103.87	7.697	4.212	409.52	1.0158
11	33	3169.43	6.533	1.995	236.89	0.9700
12	34	3185.97	7.454	5.049	465.09	0.9635
13	37	3267.73	7.229	15.553	858.18	1.0285
14	40	3356.56	6.493	2.676	274.98	0.9457
15	42	3413.79	7.454	33.110	1391.06	2.2594
16	43	3437.34	7.214	32.326	1312.81	2.4738
17	44	3487.44	7.841	2.499	324.69	0.9224
18	62	4218.67	7.260	114.536	3218.22	1.2410
19	63	4275.82	7.143	3.819	374.26	1.1468
20	66	4406.75	7.180	36.244	1423.68	1.0251
21	68	4675.57	6.268	4.584	361.55	0.9400
22	71	4810.38	7.681	30.423	1371.46	1.2252
23	74	4948.98	7.493	10.424	659.76	1.3704
24	92	5920.51	8.185	197.219	6472.33	1.1953
25	94	6018.55	8.044	215.268	6387.02	1.2742
26	105	7058.16	7.831	1.085	200.12	1.2501
27	107	7243.92	6.610	1.581	227.24	1.1031
28	108	7278.85	8.342	63.039	2726.16	1.1735
29	109	7367.69	8.349	4.054	496.46	1.0054
30	117	8885.79	9.410	4.197	228.50	1.1173
31	119	9298.30	9.995	23.311	954.29	1.1738

RUN NO = 701 NUMBER OF CHANNELS = 4095
CHANNEL NUMBER SLOPE CRITERION (IMAX) = 150
ERROR CRITERION FOR PEAK ANALYSIS (PER-CENT) 20.0
SLOPE CRITERION FOR BACKGROUND IN UNITS OF SQRT OF BACKGROUND = 0.50
PEAK LOCATION CRITERION IN UNITS OF SQRT OF BACKGROUND (BGER) = 2.0
ERROR TERM REDUCTION DUE TO SMOOTHING 1.690
LINEARITY RUN NUMBER 393 N2 = 20 N3 = 80 XN = 10.
FIRST PEAK FWHM (KEV) FPS = 7.20 FWHM VARIATION (KEV) ERFW = 3.00
ENERGY PER CH.NC (KEV) = 2.145
ENERGY-1 = 2223.3 CHANNEL NO = 563 TRUE PEAK CENTER = 562.6 WIDTH (KEV) = 7.06
ENERGY-2 = 7367.7 CHANNEL NO = 2963 TRUE PEAK CENTER = 2961.0 WIDTH (KEV) = 8.35
SOLID ANGLE RADIANS = 0.5760E-05 FLUXT = 0.6000E 12
EFFICIENCY DATA INITIAL ENERGY (KEV) = 1000. DELTA ENERGY (KEV) = 500.
0.100E-05 0.660E-04 0.220E-03 0.400E-03 0.580E-03 0.760E-03 0.890E-03 0.970E-03 0.970E-03 0.960E-03
0.920E-03 0.850E-03 0.770E-03 0.690E-03 0.610E-03 0.530E-03 0.440E-03 0.330E-03 0.220E-03
NUMBER OF PEAKS = 119

AREA(A) = SUM(COUNTS), AREA(B) = FITTED GAUSSIAN

PEAK ANALYSIS

NO.	ENERGY KEV	PK CNTR CHAN NO	HEIGHT COUNTS	H TO BG RATIO	AREA(A) COUNTS	AREA(B) COUNTS	INT(B)	ERROR(B) PERCENT	W(A) KEV	W(B) KEV	BASE CHAN	TYPE		
1	1533.2	244.1	477.6	0.493	1513.2	1813.0	6.96	7.69	6.72	7.13	7.	S		
2	1613.6	280.5	578.2	0.679	2294.1	2193.3	6.43	6.45	7.21	7.13	12.	*S		
3	1725.1	331.5	926.1	1.248	3797.0	3511.1	7.69	4.07	7.13	7.13	20.	D	0.00	
4	1810.6	370.9	161.4	0.223	405.8	611.7	1.12	18.24	5.49	7.12	5.	S		
5	2129.6	518.7	192.6	0.216	490.1	729.7	0.79	17.31	5.43	7.12	5.	*S		
NUMBER OF MULTIPLE PEAKS = 5														
THE 3 STRONGEST PEAKS ARE ANALYSED AS A TRIPLET. CHECK BACKGROUND SUBTRACTED DATA.														
6	2223.3	562.4	3793.4	4.091	13910.0	14375.3	13.85	1.59	7.12	7.12	29.	T	0.02	0.02
7	2214.9	558.5	645.6	0.695	2362.9	2446.6	2.38	5.84	7.12	7.12	29.	T	0.02	0.02
8	2205.3	554.0	264.1	0.284	998.7	1000.8	0.99	12.87	7.12	7.12	29.	T	0.02	0.02
9	2470.2	677.3	212.8	0.249	606.0	807.4	0.60	15.48	5.93	7.13	6.	*S		
10	2527.6	704.1	165.4	0.192	687.3	627.8	0.44	19.29	9.30	7.14	7.	S		
11	2682.3	776.4	250.6	0.271	839.8	953.0	0.59	13.82	7.20	7.15	7.	*S		
12	2721.5	794.7	736.1	0.740	2517.4	2800.0	1.69	5.36	7.15	7.15	10.	D	0.00	
13	2835.2	847.8	406.3	0.407	1545.7	1547.9	0.86	9.05	7.16	7.16	10.	*S		
14	2873.7	865.7	225.0	0.236	875.0	857.8	0.46	15.34	7.02	7.16	9.	*S		
15	2955.5	903.8	254.2	0.271	1067.1	970.0	0.50	13.39	7.17	7.17	14.	D	0.00	
16	3103.9	973.1	409.5	0.407	1589.1	1566.8	0.73	8.95	7.70	7.19	9.	*S		
17	3169.4	1003.7	236.9	0.225	744.9	907.4	0.41	15.33	6.53	7.20	7.	*S		
18	3186.0	1011.4	465.1	0.435	1657.7	1782.2	0.79	8.44	7.45	7.20	8.	*S		
19	3225.8	1030.0	182.3	0.163	424.2	699.0	0.30	19.76	5.35	7.21	4.	S		
20	3267.7	1049.7	858.2	0.784	3166.2	3293.8	1.40	4.96	7.23	7.21	10.	*S		
21	3292.5	1061.3	200.2	0.188	847.1	768.9	0.32	17.68	8.67	7.22	9.	S		
22	3356.6	1091.2	275.0	0.257	838.0	1057.4	0.43	13.31	6.49	7.23	6.	*S		
23	3413.8	1117.9	1391.1	1.291	5656.3	5356.7	2.12	3.29	7.24	7.24	25.	T	0.00	0.00
24	3437.3	1128.9	1312.8	1.222	5332.8	5055.4	1.98	3.44	7.24	7.24	25.	T	0.00	0.00
25	3487.4	1152.3	324.7	0.305	1165.4	1252.4	0.48	11.41	7.84	7.25	7.	*S		
26	3778.0	1288.1	181.3	0.184	634.6	704.7	0.24	18.75	7.15	7.31	7.	S		
27	3844.3	1319.1	280.5	0.253	917.7	1092.5	0.37	13.13	7.32	7.32	12.	D	0.00	
28	3854.9	1324.1	1008.8	0.915	3304.1	3929.4	1.33	4.27	7.32	7.32	12.	D	0.00	
29	4012.6	1397.7	310.2	0.304	1127.5	1214.3	0.39	11.55	7.36	7.36	12.	D	0.02	
30	4218.7	1494.0	3218.2	2.861	13700.2	12680.0	3.92	1.82	7.41	7.41	21.	D	0.00	
31	4275.8	1520.7	374.3	0.350	1521.4	1478.1	0.45	10.13	7.14	7.42	12.	*S		
32	4406.7	1581.8	1423.7	1.291	5199.5	5649.4	1.70	3.27	7.18	7.46	11.	*S		
33	4462.3	1607.8	409.4	0.387	2240.1	1628.0	0.49	9.11	10.67	7.47	11.	S		
34	4675.6	1707.5	361.6	0.359	1057.2	1449.8	0.43	10.54	6.27	7.54	6.	*S		
35	4810.4	1770.4	1371.5	1.363	5919.9	5528.3	1.65	3.27	7.58	7.58	20.	D	0.00	
36	4949.0	1835.1	659.8	0.676	2742.5	2675.3	0.80	5.83	7.62	7.62	16.	D	0.00	
37	5920.5	2288.5	6472.3	4.498	30520.4	27500.8	8.57	1.20	7.99	7.99	23.	D	0.00	
38	6018.6	2334.2	6387.0	5.055	31980.9	27270.9	8.60	1.19	8.03	8.03	30.	D	0.00	
39	6380.6	2503.0	588.5	0.778	2025.4	2565.6	0.86	5.89	8.19	8.19	12.	D	0.05	
40	7058.2	2818.6	200.1	0.208	972.2	907.7	0.35	17.34	7.83	8.53	11.	*S		
41	7243.9	2905.1	227.2	0.179	822.2	1042.3	0.41	17.01	6.61	8.62	8.	*S		
42	7278.8	2921.4	2726.2	2.081	13242.6	12531.6	5.00	2.14	8.34	8.64	19.	*S		
43	7367.7	2962.7	496.5	0.381	2068.1	2294.5	0.93	8.41	8.35	8.69	11.	*S		
44	7590.1	3066.3	352.0	0.209	1877.3	1652.6	0.71	12.73	8.83	8.83	45.	T	0.00	0.00
45	7631.4	3085.4	13818.3	8.851	72478.6	64883.0	28.06	0.77	8.83	8.83	45.	T	0.00	0.00
46	7645.5	3092.0	11091.3	7.386	59442.1	52078.6	22.60	0.87	8.83	8.83	45.	T	0.00	0.00
47	7915.2	3217.2	74.2	0.464	223.5	355.0	0.16	19.74	6.96	8.99	5.	S		
48	8885.8	3668.0	228.5	1.903	1192.2	1163.0	0.73	7.34	9.41	9.57	12.	*S		
49	9298.3	3859.2	954.3	8.617	5243.9	4985.4	3.85	2.93	9.82	9.82	22.	D	0.00	

APPENDIX D
Correction for System Nonlinearity

In order to correct for the nonlinearity of the data collecting system (gamma detector, amplifier and multichannel analyser) a method described by Heath, et al. (11) is used. A high precision dc voltage standard (0-10 volt output) with an absolute accuracy of 1 part in 10^4 is used in conjunction with a mercury relay pulser to perform the linearity check. Test pulses are inserted into the 4096 channel analyser through the preamplifer with a uniform voltage spacing. The choice of spacing depends on the gain settings of the system.

Usually 100 equally spaced peaks are put into the analyser. Two peaks, one near the low energy end and the other near the high energy end of the spectrum, are assumed to have zero linearity correction factor and the remaining peaks are fitted linearly to those two standard peaks. The standard peaks chosen could be, for example, the twentieth and the eightieth peaks. The actual peak center at channel C compared with the predicted peak center for linearity at the same channel gives the correction factor for the nonlinearity of the system at channel C.

Subroutine LINEAR calculates these correction factors and stores them in array CORR, it also needs an array CNTR to store the peak center at which the correction factor applies.

Subroutine ADJUST performs the linearity correction to a given peak center, PCNR, by use of a second order interpolation using the CNTR and CORR arrays to determine the true peak center, TPC.

It is possible to bypass the linearity correction calculation by either deleting the subroutines from the program, which means rewriting parts of the program, or by reading in linearity correction factors which are all zero. This can be done by appropriately arranging the input deck, see Appendix B.

APPENDIX E

Background Calculation

Subroutine BAKSUB performs the background calculation and subtraction on the spectra. Basically the program finds the peaks by using a first difference calculation and sets the background as a linear fit from minima to minima of the peak. However in the case of partially resolved peaks the minima must be excluded. To do this, a slope criteria is established which sets a maximum value for the slope of the background. When the angle between the slopes of the background of two adjacent peaks is greater than the slope criterion the pair will be considered as a multiplet. This procedure is repeated for up to a maximum of five peaks. The slope criteria is expressed as DCR in the input list and is multiplied by the difference in channel numbers between two minima and by the square root of the background at the upper minima.

To make the minima of a peak more representative of the true background, the counts in the two channels, one on each side of the minima, are added to the minima counts, and the average is taken. New minima are sought using this averaged value, and it is these new minima from which the linear background fit under the peak is calculated.

BAKSUB also prints out the lower channel number of each peak, the number of counts in the lower and upper channel numbers, the slope between the two minima of a peak, and the width of the peak in channels.

APPENDIX F

Intensity Calculation

Subroutine INTSTY calculates the intensity of a gamma ray at energy E after being given its area as calculated in PKANAL. There are several options open to the user in converting from the area in counts to some other quantity. Basically as INTSTY is written the area is divided by two constants, SANGLE and FLUXT, and also by a quantity GX which varies with the energy of the gamma peak being considered.

For thermal neutron capture gamma spectra, the number of gamma rays of energy E per 100 neutrons captured is the quantity of interest. Thus the factor SANGLE represents the solid angle geometry factor for the detector as well as any minor correction factors; FLUXT represents the total number of neutron captures in the sample divided by 100; and GX is the detector efficiency at energy E as obtained by second order energy interpolation of data in the EFFCY array.

The parameters SANGLE, FLUXT, and EFFCY are inputed to the program and are determined by previous calculations.

References

1. V. J. Orphan and N. C. Rasmussen, "A Ge(Li) Spectrometer for Studying Neutron Capture Gamma Rays", Nucl. Instr. and Methods, 48, No. 2 (1967).

2. Application of Computers to Nuclear and Radiochemistry, National Academy of Sciences, National Research Council - NAS-NS 3107 (1962).

3. R. G. Helmer, et al., "Photopeak Analysis Program for Photon Energy and Intensity Determinations", Nucl. Instr. and Methods, 57 (1967).

4. M. A. Mariscotti, "A Method for Automatic Identification of Peaks in the Presence of Background and Its Application to Spectrum Analysis" Nucl. Instr. and Methods, 50 (1967) pp. 309-320.

5. T. Inouye and N. C. Rasmussen, "A Computer Method for the Analysis of Complex Gamma-Ray Spectra", Trans. of the ANS 10, 1, p.38 (1967).

6. T. Inouye, "The Super Resolution of Gamma-Ray Spectrum", Nucl. Instr. and Methods, 30, (1964).

7. J. Hamawi, Ph.D. thesis, M.I.T. Dept. of Nucl. Engr. (To be published).

8. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comput., 19, 297 (1965).

9. Program FOURT available from IBM Program Information Department, 40 Sawmill River Road, Hawthorn, New York 10532

10. N. C. Rasmussen, Y. Hukai, T. Inouye and V. J. Orphan "Thermal Neutron Capture γ-Ray Spectra of the Elements", MITNE-85 (prepublication)

11. R. Heath, et al., IEEE Trans. on Nuclear Science, NS-13, 3, (June 1966) 445-456.